

2018

Augmented Reality as a Potential Tool for Filmmaking

Paul Blachfield

Technological University Dublin

Follow this and additional works at: <https://arrow.tudublin.ie/scschcomdis>

 Part of the [Computer Sciences Commons](#)

Recommended Citation

Blanchfield, Paul (2018). *Augmented reality as a potential tool for filmmaking*. Masters dissertation, DIT, 2018.

This Dissertation is brought to you for free and open access by the School of Computing at ARROW@TU Dublin. It has been accepted for inclusion in Dissertations by an authorized administrator of ARROW@TU Dublin. For more information, please contact yvonne.desmond@tudublin.ie, arrow.admin@tudublin.ie, brian.widdis@tudublin.ie.



This work is licensed under a [Creative Commons Attribution-NonCommercial-Share Alike 3.0 License](#)

Augmented Reality as a potential tool for filmmaking



Paul Blanchfield

A dissertation submitted in partial fulfilment of the requirements of
Dublin Institute of Technology for the degree of
M.Sc. in Computing (Advanced Software Development)

Date: September 2018

Declaration

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Stream), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: Paul Blanchfield

Date: September 2018

Abstract

Augmented Reality (AR) has been used for a wide variety of industries. The purpose of this study was to determine the suitability of this technology for use in filmmaking. One of the problems on a film set is the time taken to block a scene. *Blocking* involves the placement of subjects and props within a scene. Different ideas have been used for blocking including previzualisation and *Virtual Reality (VR)*. This study proposed the use of AR as a tool to solve this problem.

Marker-based and *Markerless* AR were assessed in turn to determine their suitability for addressing the problem. The use of *AR markers* and *QR codes* were examined in comparison with the use of *Simultaneous Localization and Mapping (SLAM)* implementations. The marker-based AR requires a physical object to scan and markerless is done via the mapping of GPS coordinates. Experiments were conducted on the accuracy and code required for each type of AR. These involved calculating the distances from the marker and the code required to create the virtual content.

Surveys and expert interviews were conducted with filmmakers and people working in the AR industry to determine the usability and feasibility of the proposed application. This provided a qualitative approach to the technology as the acceptance of any new system is of equal importance to how it functions.

Keywords: Augmented Reality, Filmmaking, User Experience, Marker-based, Markerless, Field of View, Blocking

Acknowledgments

I would like to sincerely thank my supervisor **Andrea Curley** for her advice and support during the course of this dissertation.

I would also like to thank my family and friends for supporting me during this research project, especially my father who read every word of this dissertation.

Contents

Declaration	I
Abstract	II
Acknowledgments	III
Contents	IV
List of Figures	VIII
List of Tables	X
List of Acronyms	XI
1 Introduction	1
1.1 Background	1
1.2 Research Project/Problem	2
1.3 Research Objectives	3
1.4 Research Methodologies	3
1.5 Scope and Limitations	4
1.6 Document Outline	4
2 Literature Review	6
2.1 Introduction	6
2.2 Augmented Reality	6
2.2.1 AR Concepts	7

2.2.2	Real World Examples	9
2.2.3	AR Software Development Kits	15
2.2.4	Limitations	17
2.3	Filmmaking	18
2.3.1	Background Research	18
2.3.2	Production	19
2.3.3	AR & Filmmaking	21
2.4	User Experience	22
2.4.1	Analysis Framework and UX Theories	22
2.4.2	Prototyping	24
2.4.3	Metrics	25
2.5	Chapter Summary	26
3	Design and methodology	27
3.1	Introduction	27
3.2	Application Design	27
3.2.1	Aim	27
3.2.2	Functional Requirements	28
3.2.3	Threat Modeling	30
3.3	Prototyping	33
3.3.1	Wireframe & Low Fidelity Prototype	33
3.3.2	Medium Fidelity Prototype	35
3.3.3	High Fidelity Prototype / Proof of Concept	37
3.4	Experiment Design	37
3.4.1	User Experience Experiment	38
3.4.2	Code Experiment	39
3.4.3	Accuracy Experiment	41
3.5	Chapter Summary	44
4	Implementation and Results	46
4.1	Introduction	46

4.2	Software Used	46
4.2.1	Low and Medium Fidelity Prototypes	46
4.2.2	High Level Prototypes / Proof of Concept	47
4.2.3	Coding	49
4.3	Results	50
4.3.1	User Experience Experiment	50
4.3.2	Code Experiment	57
4.3.3	Accuracy Experiment	60
4.4	Chapter Summary	63
5	Analysis, Evaluation and Discussion	65
5.1	Introduction	65
5.2	Experimentation	66
5.2.1	Preparing the experiment	66
5.2.2	Running the experiment	69
5.3	Evaluation	73
5.3.1	Comparison, conclusion and suggestions	74
5.3.2	Strengths and limitations	75
5.4	Chapter Summary	77
6	Conclusion	78
6.1	Research Overview	78
6.2	Problem Definition	78
6.3	Design/Experimentation, Evaluation & Results	79
6.4	Contributions and Impact	79
6.5	Future Work & recommendations	80
6.6	Chapter Summary	81
A	Code	82
A.1	Experiments	82
A.1.1	Alpha Versions	82

A.1.2	Beta Versions	90
A.2	Prototypes	98
A.2.1	AR.js	98
B	User Experience Questions	99
B.1	Survey Questions & Responses	99
B.2	Expert Interviews	101
B.2.1	AR Experts	101
B.2.2	Filmmakers	101

List of Figures

1.1	Scene from Minority Report (2002)	1
2.1	Reality-Virtuality Continuum	7
2.2	FoV of Panasonic Camera	8
2.3	AR Marker	9
2.4	QR Code	9
2.5	Pokémon Go	13
2.6	Stereo Cards game environment	15
2.7	Scene from The Silence of the Lambs (1991)	18
2.8	Bruce from Jaws (1975)	19
2.9	Models used for Star Wars (1977)	19
2.10	Preproduction Organizational Chart	20
2.11	Storyboard	21
2.12	Gulfs of Execution and Evaluation	23
2.13	Models used for AR prototyping	25
3.1	Threats on a film set	31
3.2	Application Wireframe	34
3.3	Low Fidelity Prototype	35
3.4	Medium fidelity prototype	36
3.5	Application Scenario	36
3.6	Frustum of a cone	40
3.7	Qlone mat.	43

3.8	Design process mind map	44
4.1	Hiro Marker	47
4.2	Kanji Marker	47
4.3	Example of AR.js	48
4.4	Augment models	49
4.5	Model of FoV in Augment	49
4.6	Experiment Response Rate	50
4.7	Familiarity with AR	51
4.8	Use AR for work	51
4.9	Improvements with AR	52
4.10	Expert Interview Brainstorming Session	56
4.11	Code Speed Comparison	57
4.12	Code Memory Comparison	58
4.13	Beta Version Speed Comparison	59
4.14	Marker comparison	62
4.15	glasses case	63
4.16	container	63
4.17	plastic cup	63
5.1	AR Marker	68
5.2	QR Marker	68

List of Tables

1	List of Acronyms	XI
2.1	UX Framework	24
3.1	Key for Figure 3.1	31
4.1	Number of lines - Alpha Version	59
4.2	Number of lines - Beta Version	60
4.3	Marker sizes	61
4.4	Marker distances	62

List of Acronyms

AI	Artificial Intelligence
AoV	Angle of View
AR	Augmented Reality
CGI	Computer Generated Imagery
FoV	Field of View
HMD	Head Mounted Display
IFTN	Irish Film and Television Network
IMU	Inertial Measurement Unit
IOT	Internet of Things
MAR	Mobile Augmented Reality
MRLE	Mixed-Reality Learning Environment
SLAM	Simultaneous Localization and Mapping
UX	User Experience
VR	Virtual Reality
VIO	Visual Inertial Odometer
XR	Cross Reality

Table 1: List of Acronyms

Chapter 1

Introduction

1.1 Background

Augmented Reality (AR) is present on the *X Reality (Cross Reality)* spectrum. Other elements on this spectrum include *Virtual Reality (VR)* and *Mixed Reality (MR)*. Akçayır et al (2016) define AR as a real-world context which is dynamically overlaid with context-sensitive virtual information.(?, ?). Figure 1.1 shows a well known albeit fictional use of AR in the film *Minority Report*. Tom Cruise's character can manipulate content around a screen without touching it.



Figure 1.1: Scene from Minority Report (2002)

The current market trend for AR is towards educational or entertainment content. Smart devices with built-in cameras are the main tool for viewing AR content. Since

the release of *Pokémon Go* in 2016, there has been an increase in the popularity of AR. This popularity is present within the entertainment industry (?). There has been a dynamic shift in the presentation of AR through the use of *head mounted devices (HMDs)* including *smart glasses*. This is because of the reduction in the size of the hardware and the improvements in technology. These advancements created a *ubiquitous* environment for the technology which improves user experience.

1.2 Research Project/Problem

The main problem addressed was whether AR could be a tool used for blocking a film scene. *Blocking* is the filmmaking process describing the interaction between a camera and subject. There are three different subjects to be considered, actor, extra and prop. The ultimate reason for blocking is to visualise the placement of everything within a scene. For example, a camera panning through a crowded room must show certain elements. These elements are key to the story. This is also how product placement occurs on camera. The camera pans down from an advertisement to the action below. Currently blocking is a rule of thumb assumption of what the camera can see. The AD looks towards where the camera is set up and from experience determines the field of view. This can be time intensive and can cause a high level of human error. An application that could present the field of view of the camera could potentially solve this problem. In the world of filmmaking, *“time is money”* and it is important to get every shot set up in the shortest time possible.

The research question for this dissertation was:

“Can the use of a Mobile Augmented Reality (MAR) application created using marker or markerless based Augmented Reality (AR) increase the efficiency of camera and subject placement in a filmed scene?”

1.3 Research Objectives

A breakdown of the research objectives for this dissertation are as follows:

- Review the existing literature relating to augmented reality, user experience and filmmaking.
- Create multiple levels of prototypes for an AR application.
- Prepare experiments to test the different components of the application. The three components were *Accuracy*, *Code*, and *User experience (UX)*.
- Evaluate the results of the experiments and perform an analysis.
- Identify potential research areas for future analysis.

1.4 Research Methodologies

The research methods were a mixture of qualitative and quantitative. This blend allowed for the analysis of the usability and efficiency of the technology. The quantitative data required was the distance required to view virtual content.

Secondary research was the main component of this study. This research was an analysis of the existing material written about AR. This material included research papers and software documentation. The documentation presented the best possible avenues for experimentation within the technology.

Primary research complemented the secondary research to create a more rounded study. This primary research was in the form of surveys and expert interviews. The surveys presented a view of the potential user base. The expert interviews allowed for a deeper understanding of the subject matter and problem. The surveys focused on the Irish film industry. The expert interviews introduced the AR industry into the mix.

1.5 Scope and Limitations

The scope of this research was the design and usability testing of AR applications. This was determined by using of the technology to complete a specific task. The technology was analysed on a smartphone.

The hardware used presented its own limitations. Devices such as *HMDs* reside outside of the remit for this dissertation but could be analysed as part of future research.

1.6 Document Outline

This dissertation breaks down into the following chapters:

- **Chapter 2** was an analysis of the existing literature relating to AR. A background of the filmmaking process was addressed. The concept of blocking was established as a part of this background and presented in relation to the overall problem. There was some film theory included to justify decisions made during a production. To solve the problem, user experience theories were used for evaluation. This chapter introduced the relevant theories and demonstrated the metrics which were evaluated. Gaps and commonalities between AR, filmmaking and user experience were presented.
- **Chapter 3** described the design process which was chosen for this dissertation. For this research project, prototypes were created to demonstrate the proof of concept of the application. Three experiments were conducted for each of the components being assessed, *Code*, *Accuracy* and *User Experience (UX)*. Within each experiment section the metrics for measurement were presented and justified.
- **Chapter 4** demonstrated the creation of the prototypes and the results that were gathered from the experiments. The results gathered from each experiment were compared. The languages and software packages that were chosen to conduct

the experiments were presented as well as the product created from each. The order of the experiments was *UX*, *Code* and finally *Accuracy*. The *Accuracy* experiment was further broken down into *marker* and *markerless* experiments.

- **Chapter 5** analysed the results from the experiments. It also evaluated the design process as a whole for the dissertation. Any problems encountered and adjustments made were justified.
- **Chapter 6** presented a conclusion to the overall thesis. Any contributions made to the research of augmented reality within the context of filmmaking were presented. All of the limitations encountered during the process of writing the thesis were presented and addressed. Finally there was an analysis of future work in the area of augmented reality and filmmaking which was outside the scope of this particular research project.

Chapter 2

Literature Review

2.1 Introduction

There are three parts to this literature review.

1. A background to AR technology
2. An introduction to filmmaking concepts
3. An analysis of user experience.

2.2 Augmented Reality

AR was first coined by Tom Caudell and David Mizell in 1992. It made an appearance on Milgram's *Reality-Virtuality Continuum* shown in Figure 2.1. AR was first developed by Ivan Sutherland in 1968 in tandem with the first VR system. Other key events in AR's history included the inventions of the tablet (1972) and the laptop computer (1982). (?, ?)

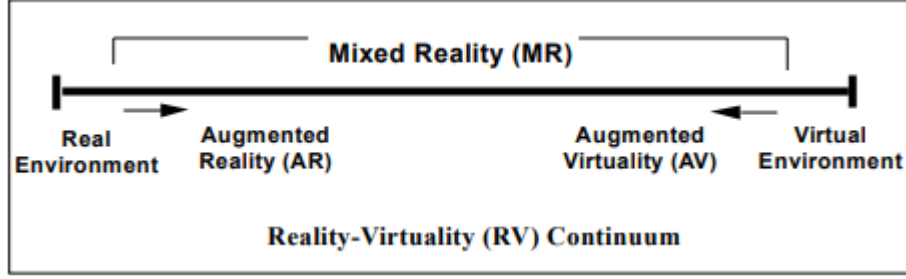


Figure 2.1: Reality-Virtuality Continuum

(?, ?)

AR is the process of placing virtual information on top of a real-world environment to “*augment*” the user’s surroundings. This is through the use of a mobile device such as a smartphone or a *HMD* (?, ?, ?, ?). Examples of *HMDs* include the *Microsoft HoloLens* (?, ?) and *Google Glass*. (?, ?).

Advancements in technology have benefited the creation of AR. These include *GPS* (1993), *markers* (1999) and *real-time feature tracking* (2006). The release of *ARToolKit* in 1999 boosted AR development. This spawned numerous *software development kits* (*SDK*). These *SDKs* included *ARCore* and *ARKit* (?, ?).

According to Liu, the goal of AR is to combine the real and virtual in a photorealistic way. This makes it difficult to distinguish between the two (?, ?). The techniques used can range from *tracking and registering features of objects* (?, ?) to using *projection mapping*.(?, ?).

2.2.1 AR Concepts

There are two concepts for the development of *mobile AR applications* (*MAR*). These are *Marker-based* and *Markerless*. The *Field of View* (*FoV*) is relevant to both of these concepts. The dictionary definition for *FoV* is “*the angle between two rays passing through the perspective center (near nodal point) of a camera lens to the two*

opposite sides of the format.¹ The *FoV* for a camera changes depending on the size of the lens. Changing the lens changes the view of the camera as shown in Figure 2.2.

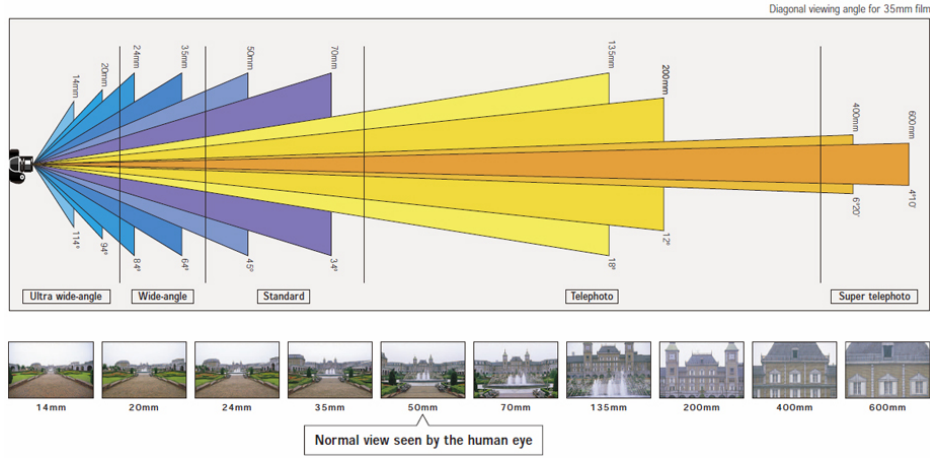


Figure 2.2: FoV of Panasonic Camera

The first step for calculating the *FoV* requires finding the *angle of view* (*AoV*). Sensor width and focal length of the camera are used in 2.1 to calculate the *AoV*³.

$$AoV = 2ARCTAN(Sensorwidth/(2 * focallength)) * (180/\pi) \quad (2.1)$$

Using the result of Equation 2.1, the *FoV*⁴ in Equation 2.2 can be calculated.

$$FoV = 2(TAN(AngleofView/2) * DistancetoSubject) \quad (2.2)$$

Markers

There are two types of markers used to present virtual information. These are *AR markers* and *QR codes*. *AR markers* are black and white and are only able to track and identify a virtual object (?, ?). *QR codes* are square barcodes that store positioning information. Other forms of information are also embedded in the code (?, ?). *QR codes* appear in advertising and have become ubiquitous within society. They often appear on tickets and receipts. *AR codes* appeared as a part of educational courses (?, ?). Figures 2.3 and 2.4 are examples of *AR markers* and *QR codes*.

¹<https://www.thefreedictionary.com/field+of+view>

³<https://shuttermuse.com/calculate-field-of-view-camera-lens/>

⁴<https://shuttermuse.com/calculate-field-of-view-camera-lens/>

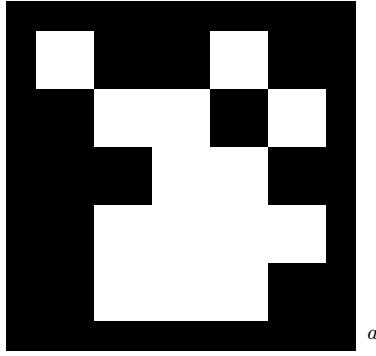


Figure 2.3: AR Marker

^a<https://pypi.org/project/ar-markers/>

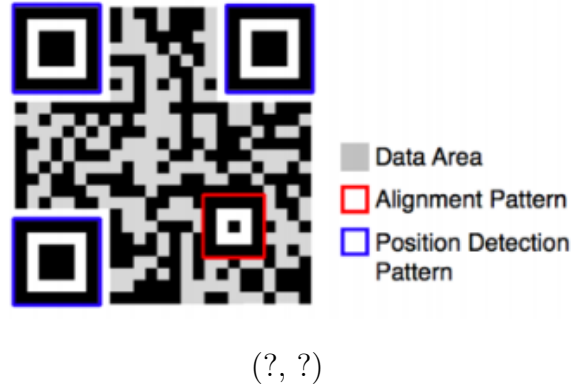


Figure 2.4: QR Code

Markerless

Markerless is the presentation of virtual imagery without the use of either *AR markers* or *QR codes*. There is a separate process which determines this position of the virtual object known as *Simultaneous Localization and Mapping (SLAM)*. It is possible to globally align a virtual object in a real scene using *SLAM*. (? , ?). For correct initialization there needs to be a baseline established. This is determined by camera pose (? , ?). Markers set limits to the visibility of AR. Using *SLAM*, a room can be scanned and objects placed at specific GPS coordinates. For example, objects placed around the room retain their positions even after the camera has moved. When you pan the camera around the room the different items will move on and off the screen. *Pokémon Go* uses *SLAM* as a part of its game mechanics. This allows developers to release content at a given location at a set time. These releases have often caused crowds to descend on the selected areas.

2.2.2 Real World Examples

This section analyses the real world applications of AR. Since it is a broad area of research there was a choice of industries. The three chosen were *Education & Training*; *Entertainment & Marketing* and *Tourism & Navigation*. The chosen industries have some overlap in their approach to the technology. Yet, there are some unique uses

which are dependent on the industry.

Education and Training

In the realm of education there are two distinct groups of people. This is beyond the scope of the standard teacher and student. These groups are *digital nomads* (teachers) and *digital natives* (students).

There is a gap between the *digital native* and the *digital nomad*. The distinction between the two is usually centered around age. If someone is under the age of 30 they are a *digital native* while those above are *digital nomads*. A *digital native* is someone who has encountered technology for their whole life. *Digital natives* are more adept at the ever changing landscapes. *Digital nomads* are those who had a period of their lives without technology on such a wide scale. They know of a time when technology was not a driving force in everyone's lives (?, ?). Overlap between the two groups appears with older students or younger teachers.

In schools, the emphasis is on *interaction*. Universities focus on *engagement*. The use of AR is the distinction between interaction and engagement. For example, school kids used the technology as a learning aid at a botanical garden.(?, ?). University students created a “*connection between the theoretical explanations and the laboratory practices*” (?, ?). This confinement of the technology to laboratory environments is also present in (?, ?) and makes an appearance in (?, ?).

The aim of AR in education was to reduce the amount of teacher assistance required within lessons to increase productivity. AR tools are used in situations where students are learning *science, technology, engineering or mathematical (STEM)* subjects. Daas (2018) presented a study describing how AR taught students how to code (?, ?). A limitation to the use of AR by teachers is the lack of knowledge that some may have towards the technology. This could be because some of them are *digital nomads*. This limitation in combination with developers lack of creative design education expertise

leads to a stop start embracing of the technology within education systems. AR and MR environments can improve real time communication (?, ?). They can also be a part of a *Mixed Reality Learning Environment (MRLE)* (?, ?). MRLE's present both real and virtual information as educational tools (?, ?).

There are contradictions between the advantages and disadvantages of using AR for education. There are mixed feelings towards the cost of AR. This could be as a result of the hardware used. The cumbersome nature of the hardware is another disadvantage of AR. Some of the devices used such as *HMDs* can often leave the user tethered to a computer. (?, ?). An example of a tethered HMD is the *Leap Motion* headset. The *FoV* for this device is *100 degrees*. Except for the lack of movement the device is comfortable. This goes against the research but this it is on the more expensive side for devices.

Tourism & Navigation

The use of technology is ubiquitous in the realm of tourism and navigation. Since the invention of the *satnav*, drivers are looking for newer ways to detect what is ahead of them, to find where they are going and to avoid being stuck in traffic. Research is ongoing on the use of AR in relation to *collision detection* while driving. A drawback to this is the possibility of distracting the driver (?, ?). These distractions could cause an accident. When a driver's focus is on one threat another could appear. There needs to be exact specifications of what is being detected and how this information is being relayed to the user.

Currently, cars use audible beeps to detect if a collision is about to occur. This could lead to the adoption of *Audio Augmented Reality (AAR)* (?, ?). With *AAR* a noise transmits when a user enters an area. This would reduce driver distraction and allow them to adapt to situations. In the case of driverless vehicles, an opportunity arises to superimpose information about the surroundings for the passengers to view in real time. Mulloni presented the idea of AR information superimposed over real-world

environments. In the study AR is used to navigate college campuses or unfamiliar areas. (? , ?).

The creation of a unique tourist experience is a concern for historical sites. With the crowded spaces and limited artifacts visible, it can be difficult to engage tourists, who often arrive at a site, grab a selfie or two with the main attractions and move on. There has been some success with the use of VR at historical sites. An example of this is at the *Sforza Castle* in Milan, Italy. A guided tour of the castle grounds with a VR element is one of the tours available⁵. Park demonstrated the creation of AR applications for immersive tourist experiences. (? , ?)

A drawback of using AR or VR applications at historical sites could be the removal of the *tactile* and *human elements*. The human element comes in the form of the tour guide. They have usually lived in the area their whole life and provide a unique tour through the site. The tactile comes in the form of getting lost in a location and the feel of being there. When led through a site by a virtual guided tour or an audio tour the focus is on the story. The tourist feels the need to keep moving instead of absorbing the surroundings. People can reject predefined narratives preferring to make smaller individual decisions. There is a market for this with bespoke experiences or *sandbox* gaming environments. A sandbox environment allows the user to explore an area on their own.

Entertainment and Marketing

The entertainment and marketing industries are a growing player in the adoption of AR. Entertainment's use of AR is in two types, *passive* and *engaged*. Going to the source of the AR determines the type. Musicians and filmmakers use passive AR to add extra elements to their work. Examples of *passive* AR include applications created by bands such as *Gorillaz* and *U2*. Gorillaz created an AR application called *Lenz*

⁵<https://beyondthegate.io/en/homepage/>

ahead of the release of their latest album *Humanz*⁶. U2 incorporated AR content as an element of their *eXPERIENCE + iNNOCENCE* world tour⁷. The users of these apps do not need to do anything other than open the app at the right time to view the material.

AR gaming allows users to move from *passive* to *engaged* interactions. The most well known AR game is *Pokémon Go*. *Pokémon Go* developed by *Niantic* and released in 2016. According to Shea (2017), the game forces players to roam an area using their device's capabilities. The *Pokémon* avatars within the game are also attached to physical locations. (? , ?). The user goes to the different locations to "capture" the *Pokémon*. Since *Pokémon* are fictional versions of animals they share traits such as habitats. For example, you can find water-based *Pokémon* near a river. An image from the game is in Figure 2.5.



Figure 2.5: Pokémon Go

⁶<http://www.factmag.com/2017/04/19/gorillaz-release-lenz-another-augmented-reality-app-anticipation-humanz/>

⁷<https://next.reality.news/news/u2-takes-fans-behind-scenes-its-new-augmented-reality-tour-0184568/>

Nostalgia and ease of use caused a surge in the popularity and level of engagement of *Pokémon Go*. No additional hardware was required to play the game other than having a smart device. The application is *freemium* allowing developers to gain revenue even though the majority of players were using the application for free. *Freemium* means that the core elements of an application are free with extra content available for a small fee.

Two shortcomings of *Pokémon Go* are that it presents a physical danger to the player and also drains device battery. Despite warnings present in the game, some users injure themselves while playing. Others are victims of attack by people taking advantage of distracted players. The other and most noted shortcoming for *Pokémon Go* is the drain on the battery that it causes. While the game is played, GPS and graphical rendering are being used. Both of which can have a detrimental effect on the battery life of the device. Some users disable any AR components on the app and use it with only GPS location and a push notification for when a *Pokémon* is present.

Pokémon Go is a markerless AR game. There has been research done on the use of markers for gaming. One such example of this is the trading card game *Stereo Cards*. In this game, the playing cards themselves are the markers that store additional information for the camera (?, ?). The game environment for *Stereo Cards* is in Figure 2.6.

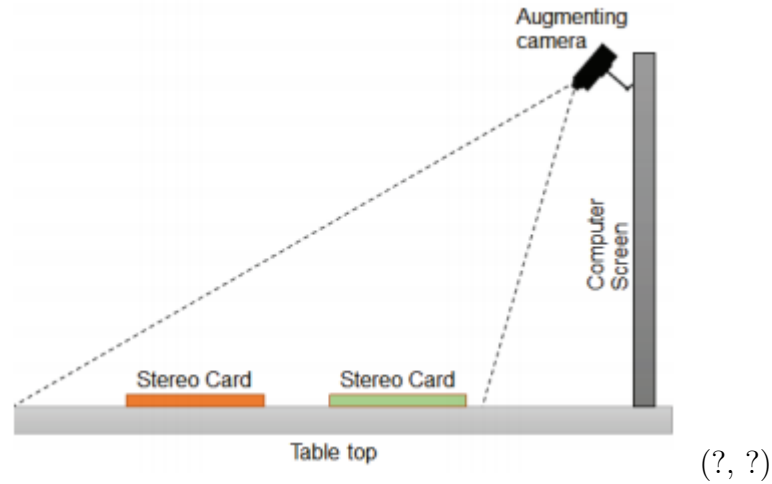


Figure 2.6: Stereo Cards game environment

Marketing corporations have tapped into the potential of AR for *gamification*. The dictionary definition of *gamification* is “*The application of typical elements of game playing to other areas of activity to encourage engagement with a product or service.*”⁹ Nazri presented an example where an AR game is available to anyone who purchases a brand of chocolate bar. To view the content the customer scans the wrapper with a smart device. The research showed that AR encourages customers to visit a company’s website. This can lead to increased sales and brand exposure. (?, ?)

2.2.3 AR Software Development Kits

The five *SDKs* discussed for of this research project were *ARCore*¹⁰, *ARKit*¹¹, *AR-Toolkit*¹², *Project Tango*¹³ and *Vuforia*¹⁴. There was an even split between the *SDKs* for *Android* and *iOS* operating systems.

⁹<https://en.oxforddictionaries.com/definition/gamification>

¹⁰<https://developers.google.com/ar/>

¹¹<https://developer.apple.com/arkit/>

¹²<https://github.com/artoolkit>

¹³<https://www.trustedreviews.com/news/what-is-project-tango-2941129>

¹⁴<https://docs.unity3d.com/Manual/vuforia-sdk-overview.html>

ARCore

ARCore is *Google's* entry into the AR development market and first appeared in March 2018. Gaming interfaces created using *ARCore* have appeared in Aultman's research (?, ?). *ARCore* requires *Android 7.0 (Nougat)* to operate. This is because of the camera capability which is part *Nougat's* source code.

ARKit

ARKit is a set of software tools released by *Apple* in June 2017. Developers used *ARKit* for creating *prototyping platforms* (?, ?) , *educational software* (?, ?) and *SLAM applications* (?, ?). *ARKit* utilises a technique called *visual inertial odometry (VIO)*. *VIO* is a combination of motion from cameras with the measurements from a *Inertial Measurement Unit (IMU)* (?, ?). *ARKit* allows for an absolute camera pose but due to a limited *FOV* can only work on special devices (?, ?).

ARToolKit & AR.js

ARToolKit is a marker based AR tool for the creation of applications. The main features of *ARToolKit* include *positioning* and *orientation tracking*. The type of AR marker is not important for *ARToolKit*. This is because it can recognise different marker patterns. Development of other AR platforms used *ARToolKit* including *AR.js*.

*AR.js*¹⁵ is a web based AR tool which runs within a browser. No external applications are required to operate *AR.js*. Using this web interface AR is visible with either a smart device or via a desktop or laptop computer. Within *AR.js* the frame rate (45, 50 or 60) is set to increase the stability dependent on the device used. Re-calibration in *AR.js* occurs when the camera moves away from the target and back towards it. Unlike *ARKit* this AR tool operates on devices older than two years old.

¹⁵<https://github.com/jeromeetienne/AR.js/blob/master/README.md>

Project Tango

Project Tango was the predecessor to *ARCore* released by *Google* in 2014. The three functions of *Project Tango* were *motion tracking*, *depth perception* and *area learning*. Motion tracking uses an *accelerometer* and a *gyroscope* to determine *position* and *orientation*. Depth perception presents the shape of the surroundings. Area learning maps out an area that the devices sees to place virtual objects within an area. *Project Tango* was only available on devices “*optimized with extra hardware - such as a barometer, motion tracking camera, and an infrared depth-sensing camera*”.¹⁶

Vuforia

Vuforia is a series of plugins and libraries for *Unity*. It was the industry predecessor to *ARCore* and *ARKit*. The main drawback of *Vuforia* packages is the cost. The cost of use per application is too expensive for applications with a small market share.

2.2.4 Limitations

The main limitation of AR is the issue of *real-time interactivity*. This means that there are no allowances for developers to define event based interactions with objects.(?, ?). As a result of this changes can’t be made to augmented graphics and sounds.

ARKit is only available on *iOS* devices released after the iPhone 6. *ARCore* relies on phones which have the *Android 7.0 Nougat*. *Nougat* is only available on phones released within the last two years. There is no way of installing it on earlier models.

A limitation with marker-based AR was the lack of room for external allowance when presenting a model. For example if the marker is a square the maximum shown is the area within the square. This renders it difficult for using a marker to establish a base point.

¹⁶<https://www.digitaltrends.com/mobile/tango-vs-arcore-theunlockr/>

2.3 Filmmaking

2.3.1 Background Research

An understanding of two filmmaking concepts are important for the context of this research. The concepts are *blocking* and *in-camera*. *In-camera* is often referred to as in-camera editing to avoid confusion with the legal definition.

Beaver defines *blocking* as an arrangement of characters and objects on a screen to create a sense of the *depth of field* and *compositional depth*. (,) The compositional depth is a more psychological reading of a scene. An example of this is if one character appears to be taller or higher on a screen than another. The viewer perceives that the taller character is more important and has a higher status in the scene. An example of this is in Figure 2.7 a scene from *Silence of the Lambs*. The idea behind the scene is to make *Jodie Foster's* character *Clarice* appear as small as possible. The director achieved this by having everyone else on screen taller than her. The idea of enclosed space comes from the lack of personal space each person has.



17

Figure 2.7: Scene from The Silence of the Lambs (1991)

In-camera editing refers to having as much visual information captured while filming as possible. This is to alleviate the need for extensive external *computer-generated imagery (CGI)* or additional editing. There is *CGI* in most filmed scenes but it is not always intrusive. The *in-camera* special effects are generally referred to as practical

effects. Two examples of this are the shark from *Jaws* also known as *Bruce* in Figure 2.8 and the models used for *Star Wars* in Figure 2.9.



18

Figure 2.8: Bruce from Jaws (1975)



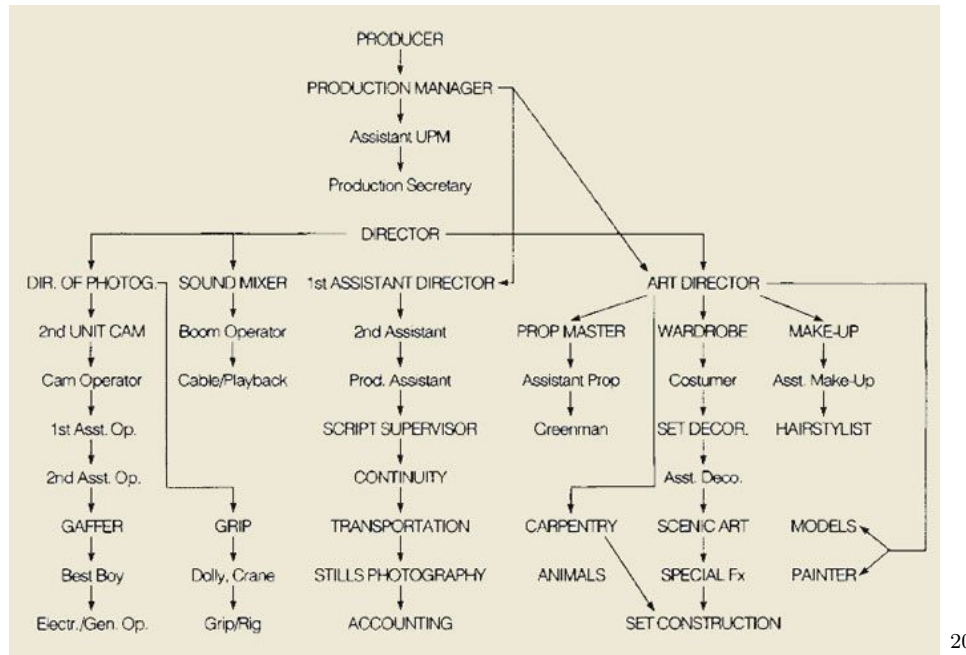
19

Figure 2.9: Models used for Star Wars (1977)

2.3.2 Production

Figure 2.9 presents an organisational chart for the preproduction of a film. The process of filmmaking is divided into three stages. The first stage is pre production which is concerned with the preparation of the different elements of the film including casting

and finding locations. Production is when the cameras are rolling and the filming is happening. The final stage is post production which involves editing so the raw footage can be moulded into a finished film.



20

Figure 2.10: Preproduction Organizational Chart

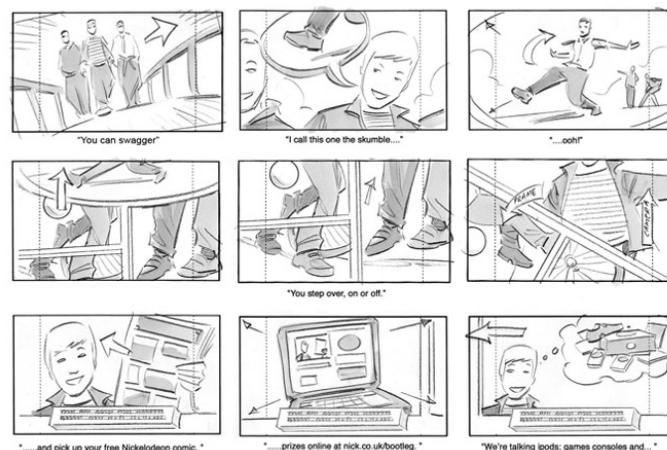
The scope of this research will focus on *directors*, *directors of photography (DP)* and *assistant directors (AD)*. These three members of the crew combined are responsible for setting up scenes and shots. A scene is an interaction between multiple actors that progresses the plot of the film. A shot could be for example showing the contents of a bag or someone walking through a door.

The director has the vision for what should happen. The DP sets up the camera and films what is happening. The ADs ensure everything in the background of a scene is where it should be to get the most out of the scene. As for the AD that role breaks down into three separate positions that branch off from each other. The 1st AD assists the director, the 2nd AD deals with the principal actors while the 3rd AD directs the background extras and other tasks. Each AD helps the one above them by taking on extra work to free up people and adjust to any situations on a film set.

2.3.3 AR & Filmmaking

There have been some examples of the use of AR within the filmmaking industry. The most prominent use of AR is the green screen alternative *ARWall*. ARWall replaces and renders a computer background in real-time with changes dependent on where the actor is moving. In traditional filmmaking this is usually done with a green screen. A green screen or sometimes a blue screen is the norm because of the ease to remove green or blue colours from an image. Other instances of AR on screen involved pre rendered computer generated avatars filmed walking around an area.

Spielman presented an idea of having an on-set editing application with a *HMD* (? , ?). This type of hardware can cause difficulty to some users without a specialized knowledge of 3D imaging. This allows the user to see *previz* information in real-time. *Previz* also known as pre-visualization is an animated render of what a shot or scene is going to look like. An animated storyboard like the one in Figure 2.11 gives filmmakers a sense of what the finished product would look like.



21

Figure 2.11: Storyboard

VR has recently become a component of the filmmaking process. This was the process for *Steven Spielberg's* film *Ready Player One* (2018). Since the majority of the film takes place within a virtual environment, the actors wore VR headsets while blocking scenes. This was to help them avoid walking straight through virtual objects while

filming.

2.4 User Experience

UX is an analysis of users and user interactions. This analysis creates a framework of who the user is and how an application can suit their needs. The UX analysis framework provides the foundation for this part of the research study. This framework was built on existing UX theories and those manipulated to suit AR needs. An analysis of prototyping and proof of concepts for AR applications concludes this section.

2.4.1 Analysis Framework and UX Theories

The design of all applications needs to comply with a set of *heuristics*. *Heuristics* are faster ways for a user to solve a problem. Aultman states that these must include user interface (UI) elements that are *easy to identify*, *clear and consistent*, easily distinguished from the surrounding environments and *contextually relevant*. The UI must always match the user's viewing angle and provide adequate feedback. (?, ?) In the research done by Pallot, the creation of a UX framework should take into account the *Quality of Service (QoS)* and *Quality of Experience (QoE)* combined with existing UX theories. (?, ?).

When creating the UX framework the first aspect to consider is how it aligns with *Norman's Seven Stages of Action Model*. The crucial elements of this model are the *Gulf of Execution* and *Gulf of Evaluation*. (?, ?). Figure 2.5 shows a graphical representation of these gulfs.

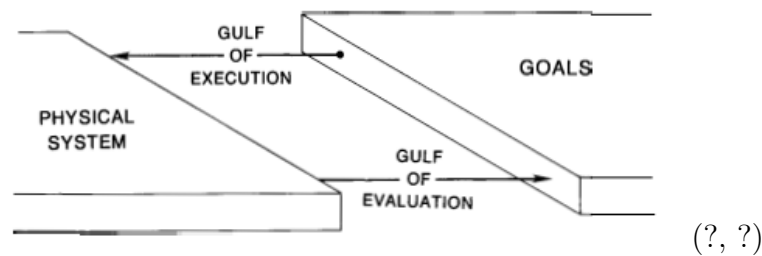


Figure 2.12: Gulfs of Execution and Evaluation

These gulfs are the gaps that can exist between the system and its goals. In the context of AR the *Gulf of Execution* would be the user was not able to present virtual content with an AR enabled device. The *Gulf of Evaluation* is when the virtual model shown is not what was expected. Other theories which were included into this framework were the *Gestalt Laws of Proximity* and *Similarity*. The *Law of Proximity* states that “items are placed near each other to appear to be a group” while the *Law of Similarity* is “similar objects will be counted as the same group”. (,). These two laws align with the concepts of AR because the idea is to present virtual content within an environment so that it appears to belong as part of the surroundings.

The research presented a way of creating such a framework by combining MAR components such as Information content with desigable elements like the ability to move 3D content (,). To conduct an evaluation it was proposed that the validity of the evaluation is clearer when multiple methods are used to confirm the result (,).

A visual representation of the UX framework from this research study is shown in Table 2.1. The column on the left contains the relevant UX theories taken into consideration. On the right there are the design components required for a MAR to pass this stage of the development. There was some duplication among the design components as they applied to numerous theories. This was inspired by existing frameworks found in the research. (,)

UX Theory	Required design components
Gulf of Evaluation	adequate feedback, user friendly UI, easy to learn, adaptability
Gulf of Execution	clear navigation, adequate feedback,
Law of Proximity	consistent navigation, grouping of elements
Law of Similarity	consistent navigation, clear navigation
Quality of Experience	adequate feedback, ability to manipulate content, easy to learn
Quality of Service	contextually relevant information, user friendly UI, adaptability

Table 2.1: UX Framework

2.4.2 Prototyping

For the creation of a *MAR* prototype three aspects need to be taken into consideration. These are ease of use(?, ?), *cognitive overload* (?, ?) and *end-user expectations*.(?, ?). The UX of an AR application is dependent on the screen size and resolution of the application as well as the potential for the use of smart glasses. (?, ?). The need for *context sensitivity* and *proactive functionality* must also be addressed. (?, ?).

It has been difficult to present AR applications in a 2D medium. To counteract this Nebeling demonstrated an alternative using modelling clay. (?, ?). Figure 2.13 shows models which were used to represent the AR content. This gave potential users tangible association with the application. A prototype is presented as a scenario of use rather than the actual presentation of the functionality.



(?, ?)

Figure 2.13: Models used for AR prototyping

2.4.3 Metrics

When the efficiency component of the research question is taken into account the following metrics needed to be collected and analysed.

- **Time:** There must be a time limit. Unlimited time would reduce efficiency of the evaluation .
- **Marker Size:** The research has shown that marker size as well as screen size are important considerations when dealing with MAR applications (?, ?).
- **Distance:** For this research the distances were calculated in feet and inches. This was because the pixel conversion were in dots per inch (dpi) which refers to the resolution of the output of an image. Equation 2.3 shows how to calculate the conversion from pixels to inches.²²

$$Pixels/DPI = Inches \quad (2.3)$$

- **UX:** This metric is concerned with the evaluation of the previous determined framework.
- **Static Analysis:** This is separate from the other metrics and happens to the source code before any compilation or execution occurs. The focus of static analysis is on syntax issues such as spacing, declaration of variables and if a variable is overwritten multiple times.

²²<https://www.quora.com/What-is-the-conversion-of-pixels-to-inches>

2.5 Chapter Summary

This chapter began with a background analysis of AR. This analysis included the real world uses for the technology. The areas discussed were *Education & Training*, *Travel & Navigation* and *Entertainment & Gaming*. Other real world examples were up for consideration but the principles remained the same. An introduction to the concepts of filmmaking gave context to the research project. This background allows for ease of presentation of the later ideas throughout the thesis. The final section of the filmmaking analysis is the current intersection of AR and filmmaking. This chapter concluded with a review and analysis of UX in relation to AR and mobile applications.

Chapter 3

Design and methodology

3.1 Introduction

This chapter acts as the software *design document* for the proposed application. A software design document is a description of the planning and implementation of an application.¹ This chapter acts as that document for the creation of any proposed applications. This document starts with an analysis of the functional requirements of AR applications. Threat modelling follows on from this with the focus on the use of applications within the confines of a film set. The different levels of prototyping were next which included the creation of high-level prototypes. The high-level prototypes acted as proofs of concept for marker-based and markerless implementations of AR. The final section of this chapter gives a detailed description of the experiment design to test the applications capabilities.

3.2 Application Design

3.2.1 Aim

The aim of the application for this research is to determine the *FoV* of a camera. This will let the user know the area seen by the camera. To achieve this a virtual

¹<http://ecomputernotes.com/software-engineering/software-design-documentation>

representation of the *FoV* appears over an area as either a model or a measurement grid.

3.2.2 Functional Requirements

With the continuing advancement of AR applications, the *Digital Manufacturing and Design Innovation Institute (DMDII)* created a list of functional requirements for AR applications. (,) The original requirements are listed in bold, with the corresponding justifications shown below them. The nine chosen functional requirements align closely with this research topic.

1. ***“The software and content generation tool shall have a user interface that can be learned by non-software literate personnel.”*** (,)
 - This is an important distinction to address because not all filmmakers are software literate. Some filmmakers are reluctant to the use of technology beyond that necessary for image capture, editing and sound.
2. ***“The software and content generation tool shall output an application that can be used by AndroidOS, iOS, and Windows OS devices.”***(,).
 - The use of *ARCore* for *Windows* and *ARKit* for *MAC* satisfy this functional requirement as it allows for use on Android and iOS devices. There is a facility in *ARKit* to make the application work across platforms. *AR.js* can operate on devices which are over two years old which is the average age of devices being used today.
3. ***“The software shall NOT require a custom version of OS, for the content to be utilized. The software shall install on a standard OS version for the smart device.”***(,).
 - As mentioned in the previous requirement there are tools available for the various operating systems. A standard OS would operate an *MAR* if the developers had utilized the correct tool in its creation.

4. *“The software and content generation tool shall allow the user to select the content by supporting all of the following: Local, Secured Server, Cloud-Based.”* (?, ?).

- Prototyping of the application offered a solution for this requirement. There needs to be a sufficient way of storing camera and lens information. Several devices using the application could link to a central location which is broadcasting the information. This central location would either be a marker or defined location on the set. There could be a cloud storage database for the different lens sizes. This database could be on a *public cloud* since the size of lenses are not likely to alter significantly between productions.

5. *“The software and content generation tool shall NOT require the use of a visual tracking tag if the AR hardware supports other tracking methods (SLAM)”*(?, ?).

- The aim of this study was to find a solution that did not require a tracking tag. A comparison between marker based and markerless AR with the aid of *SLAM* tracking forms a component of this study.

6. *“The software and content generation tool shall NOT require the use of a specific type or style of visual tracking tag.”*(?, ?).

- The rejection of this requirement could form a part of the analysis. There is the potential for using a visual tracking tag to solve the proposed problem. This would depend on the alternative methods available.

7. *“The software and content generation tool shall allow the worker to rotate 3D content.”*(?, ?).

- The term worker relates to anyone interacting with the AR application. This could be ADs or Directors. They would be able to rotate a 3D object using the application. The 3D object would be a virtual representation of the *FoV* of a camera.

8. *“The software and content generation shall support the ability for the workers to take a snapshot of their smart device (wearable and touch display) display. Allowing them to share that snapshot with others.”*(?, ?).

- Snapshots could be invaluable when recording the same scene over several days. Another use for snapshots would be in the event of jump-cut montage which is used in music videos. In previous iterations of these videos, directors used tracing paper placed over the viewfinder to track shots.

9. *“The software and content generation tool shall provide the ability to link to IoT data to specific locations by creating a ‘localization tags’ through the use of QR Code, Near Field, RFID, and/or Bluetooth tags.”*(?, ?).

- This functional requirement links to a potential use of the applications. This use would be for the creation of television shows. In the case of shows such as sitcoms there is a static number of locations. With the revolving door of directors for these shows, camera shots saved in applications like these could act as a shorthand. This shorthand would improve the workflow of the production. New directors could apply their own impression to the available camera placements with instant feedback.

3.2.3 Threat Modeling

When designing any application a threat modeling analysis must take place. The first threat to the application on a film set is the location itself. On any film set there are a lot of hazards that need to be accounted for including the amount of expensive equipment in a small confined area and the number of wires on the ground. If an application used by a member of the crew causes an accident, this could lead to a production being shut down. When a production is shut down the cost involved would be greater than the revenue saved by using the application. Figure 3.1 is a

rough estimation of the layout of a film set at an external location with the threats highlighted.

C1 \& C2	Cameras
CS	Craft Services
L1 \& L2	Lighting
P	Props
R	Road
S	Steps

Table 3.1: Key for Figure 3.1

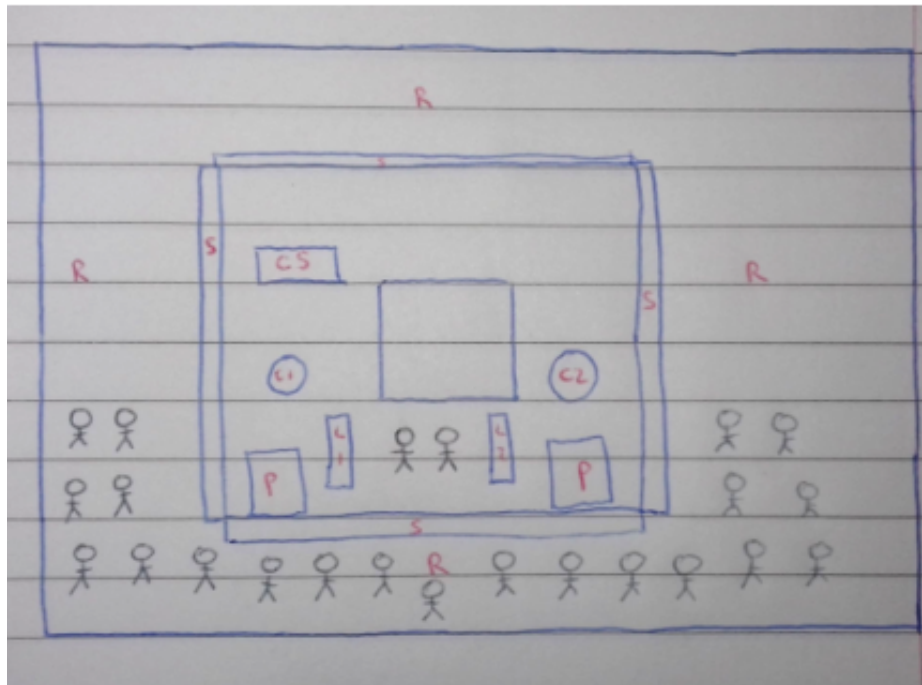


Figure 3.1: Threats on a film set

The location for the scene in Figure 3.1 is at a monument in a town centre. The first threat is the surrounding roads. The road at the top of the image remains open throughout the production. The other three roads operate on a stop/go process while scenes are being filmed. There are wires on the steps leading up to the monument. If

the film is set in but filmed in early spring, the actors and extras would be wearing coats between shots. These coats could be left on the steps causing an extra threat. There are cameras placed on either side of the monument, with a lighting rig surrounding the principal actors in the middle. A collision with any of these pieces of equipment is a major threat to the production. Some further threats which could be found in this example include scene dependent props and kettles being boiled on the craft services table.

For this scene an AR application would act as a guide for placing extras in their positions on the roads. They should be visible to the camera without causing harm to themselves or the crew. The application must not conceal any real threats. This corresponds with research that states that virtual content should not alter the real objects in such a way that is disorientates the users.(?, ?).

Another threat to the use of technology on a film set, are signals such as *WiFi* or *Bluetooth*. On a film set, all mobile devices must be switched off for several reasons. The most obvious threat being the noise created by a phone ringing. All feedback from an application must be silent or cause the device to vibrate. The other disruption caused by mobile devices is from the transmission of signals (the interference noise you hear on a radio before a phone rings). This is also the case with *Bluetooth* and could affect how the cameras record the footage. A potential solution for this would be to use the GPS settings within the device itself so the application would work while the device is operated in airplane mode.

The use of a *public cloud* to store data presents its own threats. The first being the privacy of the data stored on the service. There are also issues with the metering of the usage of a *public cloud*. It could be easy to use more data than available in the subscription. This could cost a lot of money by the end of a production. The final threat with the use of a public cloud is the loss of service. If service is lost and all of the saved snapshots are missing this could delay the blocking process. (?, ?). A

private cloud could be used to mitigate these threats providing a more secure storage location. Local storage devices could also be used.

3.3 Prototyping

Prototyping is a key step in the design of any application. The usual breakdown is to have a wireframe followed by three levels of prototypes. The wireframe remained as a 2D image. The changes to the prototypes are listed below.

- **Low Fidelity Prototype:** showed a representation of what the user would see. This is instead of the traditional *paper prototype*. A *paper prototype* is a hand-drawn visualization of what the final application would look like.²
- **Medium Fidelity Prototype:** This prototype acted as a demonstration of the use of the application with the presentation of a scenario.
- **High Fidelity Prototype:** This acted as a proof of concept for the proposed functionality of the application.

3.3.1 Wireframe & Low Fidelity Prototype

The *wireframe* was developed with the use of *Balsamiq Mockups*. The design for this application was to have two screens as shown in Figure 3.2. The circular Scan button is consistent between both screens. The screen on the left shows the virtual content after scanning an AR marker, or the selection of a model. On the right-hand side, there are three drop-down buttons.

- **Camera Type** allowed the user to select a predefined camera. Each camera had optional lenses available. Once selected the user could view the virtual model of the area visible with the camera.
- **Saved** revealed a list of previously selected camera and lens combinations. This would have worked in situations where a new director is going to an already

²<https://www.webopedia.com/TERM/P/paperprototype.html>

established location. This linked in with the functional requirements for snapshots. The snapshot in this instance is an image of the camera placement and its corresponding *FoV*.

- **Share** was for sharing the collected snapshots. This could be done by either email or other messaging service. Similar to applications like *Panascout*³ the users could write descriptions of the blocking.

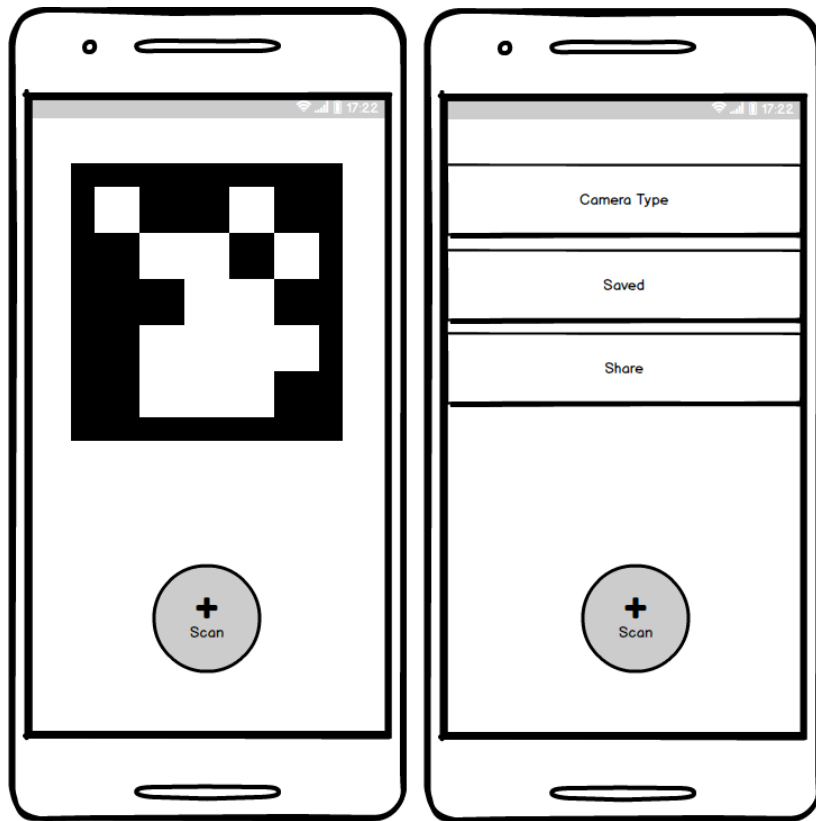


Figure 3.2: Application Wireframe

³<https://www.panavision.com/products/panascout>



Figure 3.3: Low Fidelity Prototype

Figure 3.3 is a representation of the the difference in the FoV of two cameras with the same lens. The orange line represents a *Canon 5D MKIV* while the blue line is an *ARRI Alexa LF - 16:9*. Even though there is only a slight difference between the two lines this could cause a significant difference in what was seen.

3.3.2 Medium Fidelity Prototype

The next stage of the prototype can be seen in Figure 3.4. This prototype was also created using a *Balsamiq Moqups*. This prototype was a demonstration of the navigation through the application. Figure 3.3 presents the medium fidelity prototype while Figure 3.4 was an representation of a scenario for the system.

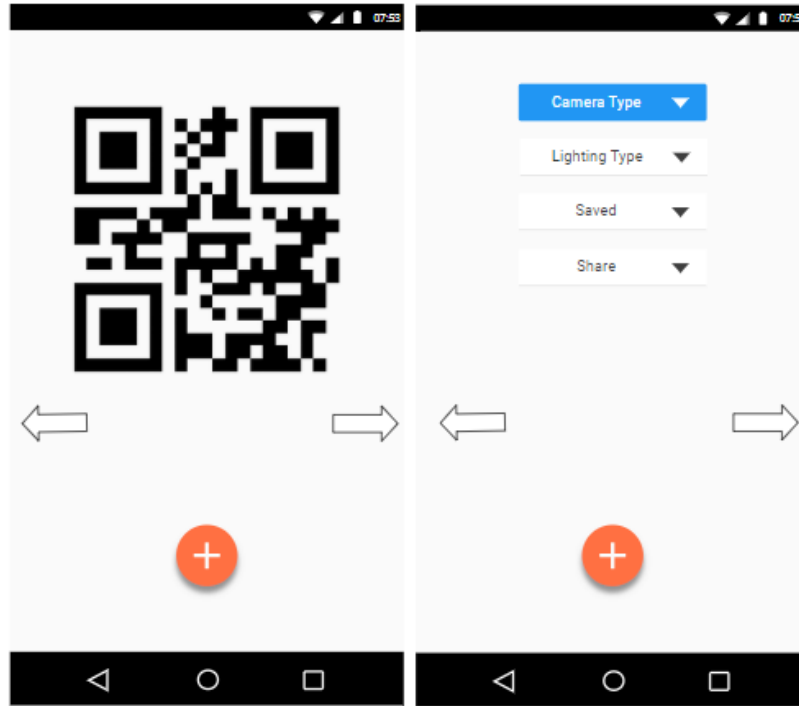


Figure 3.4: Medium fidelity prototype

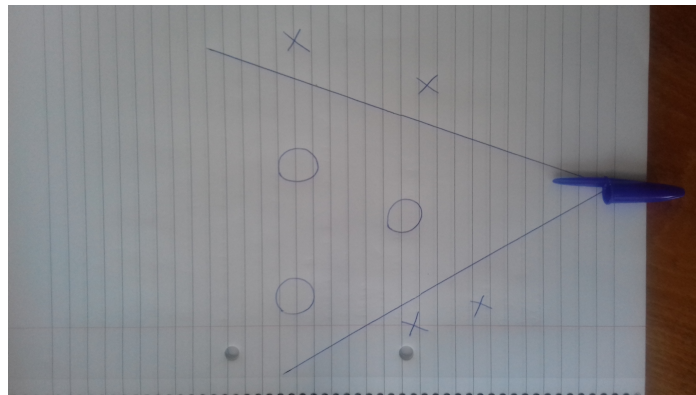


Figure 3.5: Application Scenario

The design has remained similar to the wireframe in Figure 3.2. The main differences were the additional buttons for ease of navigation between the two screens and selection of the lighting type. Lighting was included as a result of the research. There was to be a limited amount of interaction between the user and the application. This was because during the workflow on a film set there needs to be smooth transition between tasks. This led to a *“point and shoot”* style of design. In Figure 3.5 the pen lid represents

the camera with FoV visible. All the *O*s are visible by the camera camera and the *X*s are not.

3.3.3 High Fidelity Prototype / Proof of Concept

For this research the creation of two high fidelity prototypes occurred. One of these for marker based AR and the other, markerless. The first prototype was for the marker based AR application. It was created using a combination of *AR.js* and *ARToolKit*. The second prototype was a proof of concept of how markerless implementation could operate on a film set. The *Implementation and Results* chapter demonstrates the high fidelity prototype and proof of concept.

3.4 Experiment Design

Three experiments were conducted during the course of this research. Each experiment shifts focus to a different element of the study. UX was the subject of the first experiment. The second experiment analysed the source code of the application. The third experiment concerned the accuracy of marker-based and markerless AR.

The first experiment was a *qualitative analysis*, while the other two experiments were *quantitative analysis*. Qualitative analysis is an understanding of opinions and quantitative is about numbers. Acceptance of the hypotheses occurred if there was a significant increase in the efficiency of how a scene was blocked and a significant improvement to the process by the proposed application.

Various reasons could have led to the rejection of the proposed hypotheses. The main one was, if both marker and markerless implementations of AR were not suitable for solving the problem. This could have been because of the absence of available resources or the complexity of the application.

Rejection of the hypotheses would be caused by a dismissal of the idea by the experts.

This could be because of the feasibility of using the application within a working environment. If there was no engagement from the demographic it would render the technology useless within the proposed context.

3.4.1 User Experience Experiment

Stage 1: Filmmakers Survey

The first stage of this experiment focused on a single demographic. The demographic chosen was people who are currently working within the Irish film industry. The professions of these people include directors, AD, DPs, location managers and members of the art department. The art department of a film handles all the props. The source for this demographic was the the *Irish Film and Television Network (IFTN)* website crew database. This is a catalogue of people working within the industry today. This gave an overview of the perception of AR within the Irish film industry.

Stage 2: Expert Interviews

The second stage of this experiment presented the proposed application design to experts working in the fields of filmmaking and AR. On the filmmaking side, this provided a deeper analysis of the usability of the application. It expanded on the knowledge gathered from *Stage 1* of this experiment. The filmmakers critiqued the design and offered suggestions for future iterations.

AR experts provided an insight into the programming languages used and potential pitfalls in the design. This provided the researcher with a granular analysis of the technology. This got to the core of whether the application was feasible. These experts gave their opinions on the proposed solution and made suggestions for future versions.

3.4.2 Code Experiment

The code aspect of the experimentation was an *AGILE* development of the mathematical components of the application. The mathematical component is because there are *SDKs* on the market to facilitate camera permissions and computations for global positioning. *AGILE* development was the best choice for this research project because of its iterative nature.

This development was in three stages to reflect the design and testing aspects of *AGILE* development. *Stage 1* was the initial creation of the functions and tests for speed and memory consumption. *Stage 2* was the static analysis of each of the programs. *Stage 3* took the information from *Stage 2* and rewrote the programs to reflect the feedback. The tests were run again and the results compared.

Stage 1: Alpha Version

There were five calculations before the virtual object was created. Scanning a QR code triggers the creation of the object. To achieve this there were five functions created, one for each of the calculations. A sixth function will call the other five functions and return the results. The first two calculations linked with the camera itself and vary dependent on the type of lens used. Equation 3.1 and 3.2 represent the *AoV* and *FoV*.

$$AoV = 2ARCTAN(Sensorwidth/(2 * focallength)) * (180/\pi) \quad (3.1)$$

$$FoV = 2(TAN(AoV/2) * DistancetoSubject) \quad (3.2)$$

Equation 3.2 depended on the result of Equation 3.1. The *FoV* is required for the creation of a frustum. Figure 3.6 is a 3D depiction of the frustum of a cone.

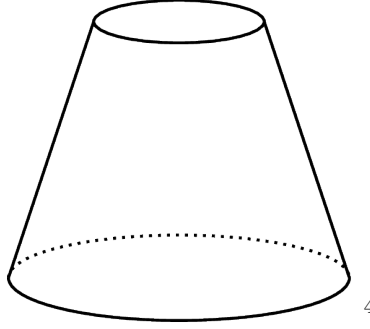


Figure 3.6: Frustum of a cone

This had the potential to be one of the most efficient ways of presenting what a camera can see. The small side at the top represents the lens of the camera while the large side is the area visible. The three equations linked with a frustum were *lateral surface area* (Equation 3.3), *surface area* (Equation 3.4) and *volume* (Equation 3.5).

$$Lat = (pi * (R + r)) * L \quad (3.3)$$

$$SA = pi(R + r) * sqrt((R - r)^2 + h^2) \quad (3.4)$$

$$V = pi/3h(R^2 + r^2 + R * r) \quad (3.5)$$

The four programming languages under review for this experiment were *C++*, *Java*, *JavaScript* and *Swift*. All four languages are object oriented.

C++ and *Swift* are the main programming languages for *ARCore* and *ARKit*. *C++* is compatible with *OS X* application development. *Swift* is only for *OS X* devices. *Java* is one of the languages at the foundations of *Android* programming. *Java 8* libraries link with *ARCore*. *JavaScript* links to AR via *AR.js* which allows devices of differing capabilities to run AR applications.

Stage 2: Static Analysis

The second stage of this experiment was a static analysis. The aim of static analysis is to analyse code without execution. This analysis is at a syntax level and concerns

the declaration of variables and indentation. Due to the nature of this analysis various software packages were required. From the research conducted the four tools chosen were:

- *CPPCheck* (C++)⁵
- *Infer* (Java)⁶
- *JSLint* (JavaScript)⁷
- *Xcode* (Swift)⁸

There are some differences with how this experiment was conducted for *Swift*. The code will be written in *Notepad++* and compiled using *Rextester* on a *Windows* computer. However, the static analysis was conducted on a *Macbook Pro*. The *Swift* was copied from *Notepad++* and analysed in Xcode which has a built in static analyser.

Stage 3: Beta Version

Following on from *Stage 2*, the static analysis results were analysed and the feedback applied to the four programs. Once these improvements were made, the speed and memory tests were run again to determine whether there was a significant difference between the two versions of the program. It is important to note here that there may not be an overall improvement as static analysis improves readability and syntax more than functional operations. The results of these tests were graphed alongside *Stage 1*.

3.4.3 Accuracy Experiment

This experiment was divided into two stages.

- **Stage 1:** focused on the use of *AR markers* and *QR codes*. With the use of these codes the distance between the marker and the device needed to be

⁵<http://cppcheck.sourceforge.net/>

⁶<http://fbinfer.com/>

⁷<https://www.jshint.com/>

⁸<http://docs.oclint.org/en/stable/intro/installation.html>

determined. The conditions for the use of Marker-based AR were considered in this experiment. This is because the available space on a film set is limited.

- **Stage 2:** turned the attention to the replication of objects which could be used for blocking a scene. These objects could be placed in different locations on a set when the shot was set up.

Stage 1: Marker-based

One hypothesis which this experiment was trying to prove was whether the distance between the source and the marker is proportional to the size of the marker required. For example if a standard marker which is $400 \times 400px$ had a maximum distance from a camera of four feet. Then if the ratio between the two was $1:1$ a marker which was $1600 \times 1600px$ would have a maximum distance of 12 feet. With the first part of the experiment the exact ratio between the marker and the source was calculated and evaluated.

The other crucial element regarding the marker experiment was the level of interference that the camera can deal with in relation to the marker itself. A marker which is unusable with any interference would be pointless for the task at hand.

All of the markers created for this experiment were $96dpi$ and as a result the corresponding markers in inches can be seen in Figure 4.14.

Stage 2: Markerless

There are alternatives to how the virtual models were presented. This was where the markerless version of AR came into play. To determine the accuracy of using markerless AR for displaying content a comparison with physical objects was required.

To achieve this objects were scanned with the applications *Qlone*⁹ and *SCANN3D*¹⁰. These applications were chosen because they represent the different methods used for

⁹<https://www.qlone.pro/>

¹⁰<https://3dscanexpert.com/scann3d-android-photogrammetry-app-review/>

scanning objects.

With *Qlone*, objects are placed on a grid like the one in Figure 3.6. This renders a 3D version of the object that can then be imported in to *Augment* for testing. *SCANN3D* uses *photogrammetry* to scan the image using multiple points.

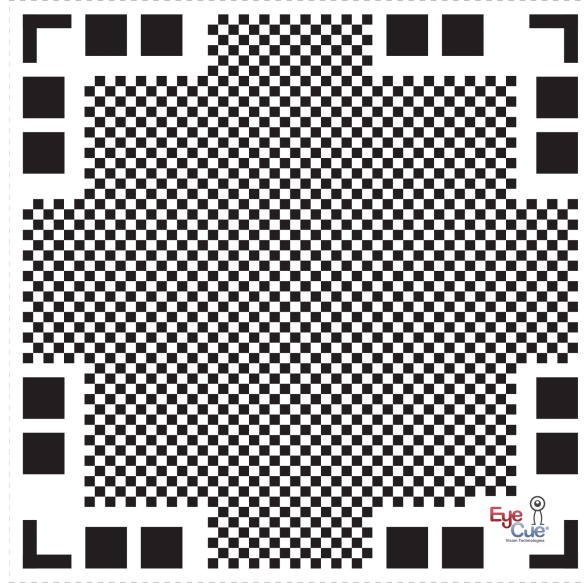


Figure 3.7: Qlone mat.

The results of this experiment were classified under three headings, *scanning time*, *rendering time* and *similarity to original*. This experiment was similar in nature to the one conducted in the research paper about the use of a *Fisheye lens*. (?). Their experiment used 3D printing and 3D modelling techniques for AR to offer a direct comparison. In contrast, this experiment used “*off the shelf*” applications. This linked with the previously defined functional requirements for a *MAR application*.

3.5 Chapter Summary

Figure 3.7 is a *mind map* of the design process for this research project. The right hand side of the map is the experimentation while the left is concerned with the prototyping of the applications. The nodes labelled red are avenues pursued during the course of this research that were dead ends. One of these was sentiment analysis which was the original *Stage 1* of the UX experiment. The idea was to gather opinions about AR and AR in filmmaking by an analysis of tweets. During this process a decision was made to write a sentiment analysis script in *Python* instead of the traditional method of using *R*. The results found proved to be insignificant for inclusion in this thesis. There was a very high level of neutral tweets which didn't provide any substantial data to analyse.

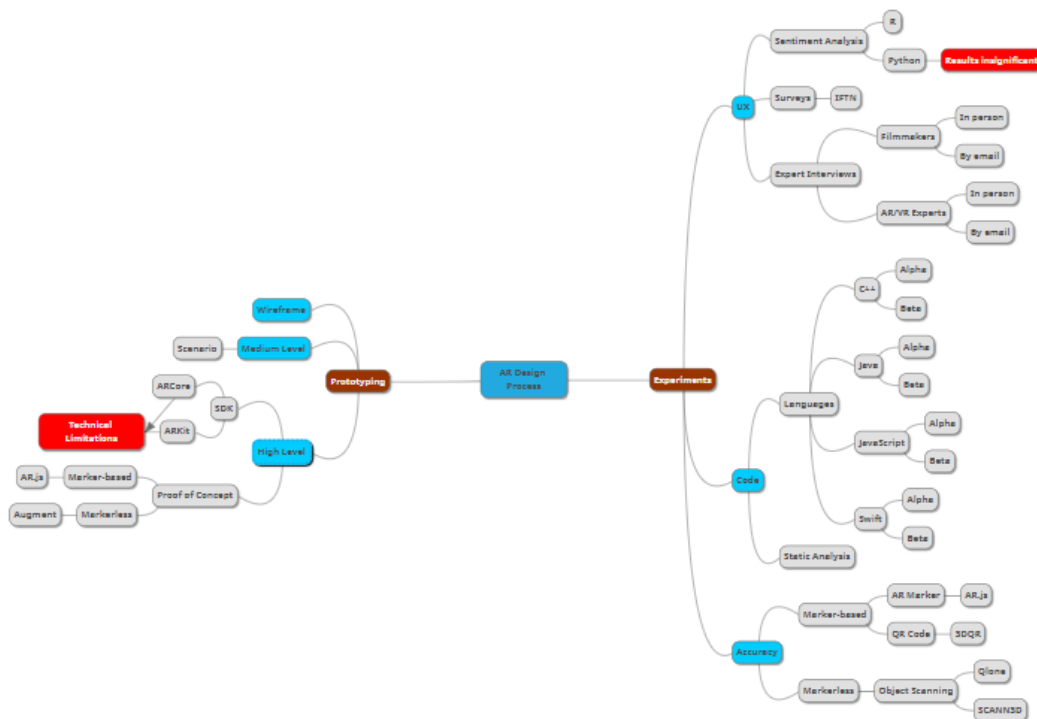


Figure 3.8: Design process mind map

The other roadblock occurred during the prototyping stage of the research. Originally prototypes were to be designed using one of the available SDKs namely *ARCore* or

ARKit. Due to the limitations of the hardware available adjustments were made to how these prototypes were conducted. The shift was towards an analysis of proof of concept using existing applications such as *AR.js* and *Augment*. Throughout the design process various decisions were made in an overall *AGILE* approach to the research. This iterative approach allowed for changes to be made “*on the fly*” while still retaining the core ideas of the thesis.

Chapter 4

Implementation and Results

4.1 Introduction

This research was implemented using a combination of programming languages together with existing applications such as *Augment*.

An explanation of the different software packages used for the prototyping and experimentation is given. The results of the experiments were also shown.

4.2 Software Used

4.2.1 Low and Medium Fidelity Prototypes

The software package used for the creation of the wireframe and medium fidelity prototypes was *Balsamiq Moqups*. *PhotoCineView* was used to present the low fidelity prototype. Alternate software packages considered in this research were *Proto.io*¹ and *UXPin*².

¹<https://proto.io/>

²<https://www.uxpin.com/>

4.2.2 High Level Prototypes / Proof of Concept

Marker based prototype

To create the marker based prototype *AR.js* was used which has an *ARToolkit* backend. The code to initialise the *AR.js* is a HTML file which was created by *Jerome Etienne* and was available from his *GitHub* page³.

AR.js allowed the camera on a device to be able to render virtual information for AR. Code written with *ARCore* or *ARKit* has a device compatibility larger than code written with *AR.js*. The device used for *AR.js* was running *Android 5.0 (Lollipop)* and was therefore not compatible with *ARCore* or *ARKit*.

The second part of the program was concerned with what markers can be used for presenting the AR. The code for this prototype can be seen in *Appendix A Figure*. Figure 4.1 shows a *Hiro* marker which was the AR marker chosen for this research study. An alternative can be seen in Figure 4.2, this type is called a *Kanji* marker. Figure 4.3 is a screenshot of what occurs when Figure 4.1 is scanned using a mobile device with *AR.js* activated.



Figure 4.1: Hiro Marker



Figure 4.2: Kanji Marker

³<https://github.com/jeromeetienne/AR.js/blob/master/README.md>

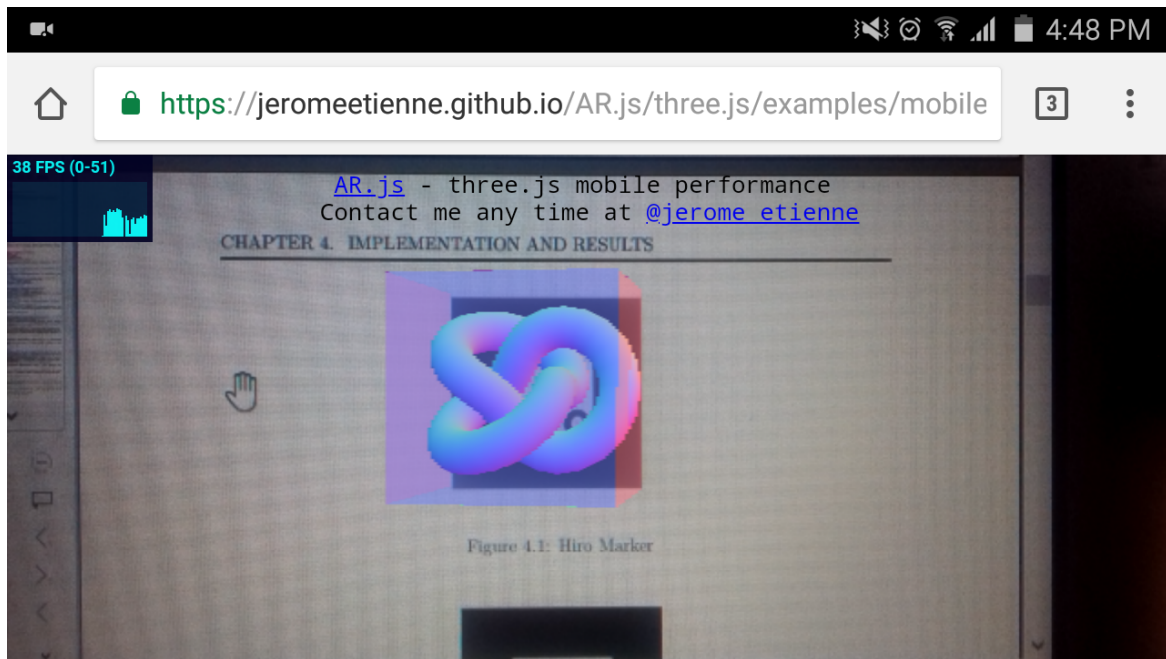


Figure 4.3: Example of AR.js

Markerless prototype

The idea behind the markerless proof of concept was the combination of multiple existing application features with additional complexity. A breakdown of the proof of concept presents was as follows. The application takes stored mathematical information which has been gathered on cameras and stored within a database either in the cloud or local storage device. For example, if the film was shot in an area where internet access was limited then stored information on the local storage device would be more efficient, even though it may slow down the speed of the application at loading time.

The application *Augment*⁴ was used for this proof on concept. After the mathematical information has been received a 3D image would be created with a variety of programming languages. This displayed the frustum of a camera. There are drawbacks to the use of this application such as the size of the models in relation to the environment.

⁴www.augment.com

Figure 4.4 is a representation of the Saved models page as shown in *Augment*. Figure 4.5 is how a model looks when application is used. In the image the edges of the model are obscured by the sides of the screen even if the device is moved. The transparency of the model is crucial because without it, the subjects to be arranged would not be visible. However, if the model is too transparent then the accuracy of the model would decrease accordingly.

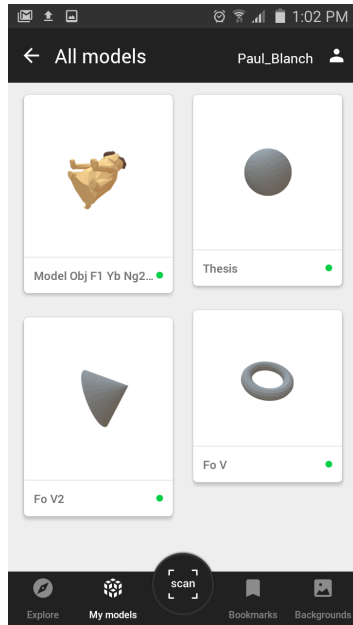


Figure 4.4: Augment models



Figure 4.5: Model of FoV in Augment

4.2.3 Coding

All of the source code was written in *Notepad ++*. The online compiler was *Rextester*⁵ which is written in *JavaScript* and can compile each of the languages. The four static analysis programs were *JSLint*, *PMD* and *CPPCheck*. *Xcode 9* has a built in static analyser for *Swift*. After the code was originally written in *Notepad++* it was copied into *Xcode* for testing.

⁵<http://rextester.com>

4.3 Results

4.3.1 User Experience Experiment

Stage 1: Filmmaker Surveys

Participants for this stage were contacted via the *IFTN* crew database. Of the 165 messages sent there were 9 replies. This was a 5.5% response rate. In Figure 4.6 the red represents the amount of people who responded to the survey. The answers from these surveys are available in *Appendix B*.

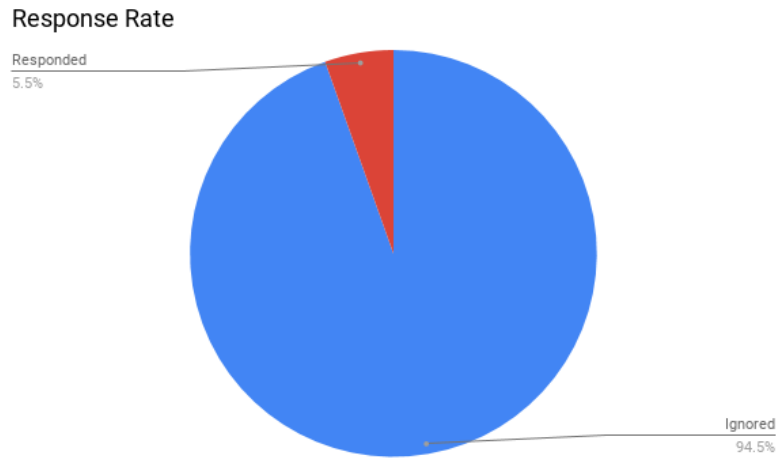


Figure 4.6: Experiment Response Rate

1. ***Are you familiar with AR and if so have you ever used an AR application?***

- (a) This question gauged the familiarity of the participants with AR. Over 40% of the participants were familiar with AR. However, not many had used the technology. In Figure 4.6 the blue segments represents the amount of people familiar with AR.

Familiarity with AR

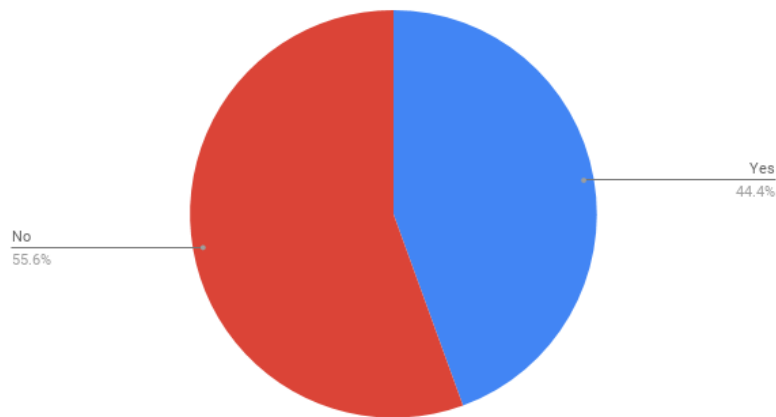


Figure 4.7: Familiarity with AR

2. *Would you ever consider using an AR application for work?*

- (a) The responses for this questions were relatively close. Those against AR were reticent about using the technology. However, those who were unsure about AR provided the most interesting insights about what to consider when using the technology. These considerations included the level of complexity and the cost. The Figure 4.8 *Yes* in in blue, *No* in red and *Maybe* is yellow.

Use AR application for work

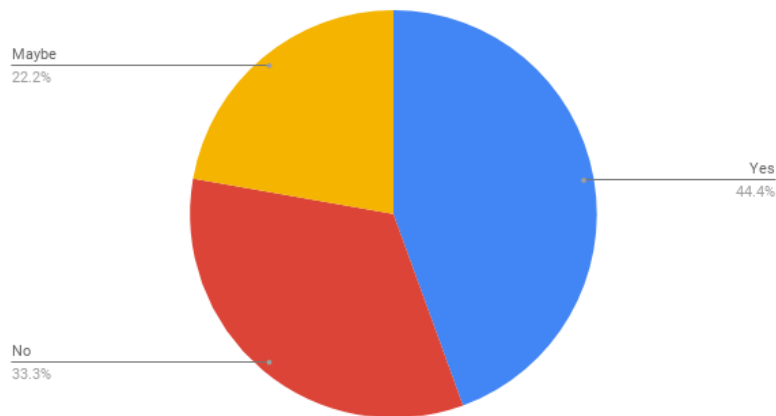


Figure 4.8: Use AR for work

3. *As part of your daily work are their aspects if any which could be improved by the use of AR?*

- (a) There were mainly positive response for this question. One participant in particular demonstrated a knowledge of the subject matter. They explained various potential uses for AR which included previsualization and how to determine the angle of the sun. Other responses included the need for specificity of shots where AR would be more efficient than traditional film-making methods. In Figure 4.9 the yellow represents *Real Time* concepts with red showing *CGI / Green screen content*. The large blue segment are the participants who couldn't determine any additional uses for AR.

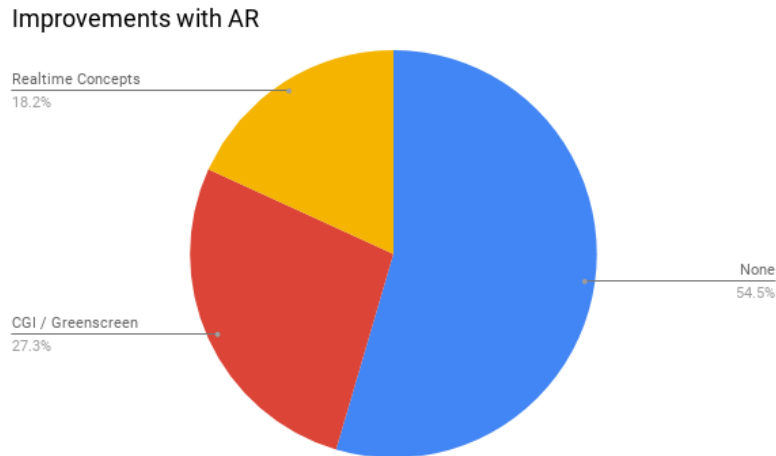


Figure 4.9: Improvements with AR

Stage 2: Expert Interviews

AR Experts

The interview questions for the AR experts were slightly different from those given to the filmmakers. The participants were *Niall Campion* from *VRAI* and *James Corbett* from *Simvirtua*. The results for this stage will be presented with (a) representing James Corbett and (b) for Niall Campion. James' answers are from an email interview while Niall was interviewed in VRAI's office.

1. *From an industry standpoint does the proposed idea sound feasible?*

- (a) *“Yes. I think it is a great idea and an appropriate application of the technology. I should mention that I have seen something like it before but that is a good thing because if you do not have competition you probably do not have a good idea! Beyond that I do not know enough about the movie making business to give a more informed opinion.”*
 - (b) Niall was in agreement about the feasibility of the proposed application.
2. ***Which programming language would you recommend for the development of the applicaiton and why?***
- (a) *“Our tool of choice for AR/VR development is Unity which means coding in C sharp. Unless you have a particular need for Unreal Engine (e.g. top level photo-realistic graphics) then I wouldd recommend Unity.”*
 - (b) The staff in VRAI mainly utilise AR SDKs which are primarily written in C++ or Swift.
3. ***Would you have a preference over the use of marker or markerless AR from a development point of view?***
- (a) *“With ARKit and ARCore we are rapidly moving away from marker based AR, and that is the way I would recommend to do it if you can. But it may be that for your particular application you will get better results using large markers. That will require lots of testing. From a user perspective markerless is to be preferred obviously.”*
 - (b) Similar to the previous answer there is a move away from using marker-based AR. The suggestion was made to use a combination of the two techniques which could achieve the best results.
4. ***Based on your knowledge and experience have you any feedback about the proposed application?***
- (a) *“As I said above, I think it is a great idea and application. And I have seen someone else doing it (I can not remember who). I can see how it solves a*

problem in a way that is better than other approaches (i.e the tech adds real value)."

- (b) The feedback was the idea of using *HMDs* such as *Microsoft HoloLens* which could easily scan a room and determine all of the surfaces. The process of renting devices as a comparison to traditional filmmaking techniques was also discussed.

5. ***Have you any recommendations for potential areas of research within AR as a whole?***

- (a) *"It is a cliché but I really do think the only limit is your imagination. In the long term at least. In the short to medium term I think there will be huge opportunities in manufacturing, construction, architecture, engineering, etc."*
- (b) Having covered most of the new research in the previous question there was a discussion about the concept of device free AR. This would be AR displayed on a screen without the need for a *HMD*. There are some prototypes available but the research is still ongoing. Other topics discussed were the use of AR for *previz* which mirrors Ichikari's research (?, ?).

Filmmakers

The filmmakers interviewed for this stage were *Flavia Pordominsky* (*Assistant Director*) and *Paul Brady* (*Director*).

Both participants answered the same questions. Flavia answered the questions via email and her answers can be seen below. Paul was interviewed in person and the contents of his interview are shown in Figure 4.10.

1. ***Are you familiar with Augmented Reality?***

- (a) *"I am familiar with Augmented Reality but not in films, I have seen it used in apps for home decoration where people can see where to place furniture in a room or to see how different paints or wallpapers look on the wall."*

2. *Would you consider using an Augmented Reality application as part of your daily work?*

- (a) *“I think that an AR app would be very useful for pre production in films as crew members will be able to see how the sets are going to look by the art dept., how many extras are going to be needed for each scene and how lighting will affect the cinematography. On production, It will be very beneficial for the art and props dept when moving elements near the camera. It would help the work of the ADs to show other dept what is going to be filmed next, in case other crew members need to be informed.”*

3. *Does the idea presented for the application make sense for its intended use?*

- (a) *“The photo does not represent much of what was explained about the app.”*

4. *What improvements would you make to the design of the application?*

- (a) *“I think it would be great if the app had a greater variety of colours as it would show better how the scenes are going to look. This will help the lighting dept. to show different moods on the film.”*

5. *What alternatives would you see for the use of Augmented Reality applications on film sets?*

- (a) *“It would be amazing if with the app we would be able to show movements and the different angles of the camera as well as showing which lights are on and off and change them just touching the screen.”*

The interview with Paul Brady took the form of a brainstorming session regarding the application and potential uses for it. Figure 4.10 shows the output of that session. On the bottom left hand corner of the image the two boxes represent different views which the application could be used for blocking a scene. There is a top down view and a side angle view. This also determined the differences in *FoV* of cameras placed

CHAPTER 4. IMPLEMENTATION AND RESULTS

beside each other. Having used AR applications for other tasks he had a high level of familiarity with the technology. Also recommendations for software to use as a part of this study can be seen including *Panascout* which was another application used for camera placement.

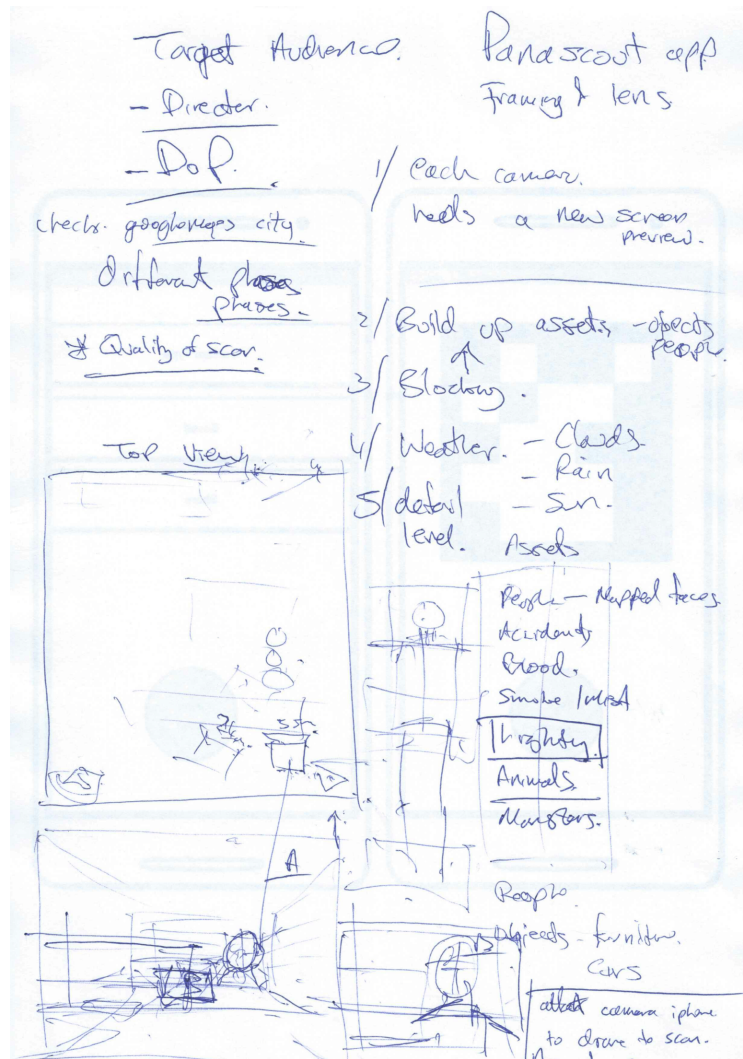


Figure 4.10: Expert Interview Brainstorming Session

4.3.2 Code Experiment

Stage 1: Alpha Versions

All four programs were run through the compiler *Rextester* ⁶. The metrics revealed by the compiler were *compilation time*, *absolute running time*, *CPU time*, *memory peak* and *absolute service time*. With the exception of memory peak all of the values are in seconds. *Rextester* allows users to change the compiled language. Each of the programs ran three times and the average results calculated. The *alpha versions* of the code are in *Appendix A*.

Each program was given the same four variables: *focal length*; *distance*; *radius* and *sensor width*. From this information, the programs were tasked with calculating the *AoV*, *FoV*, *lateral surface area*, *the surface area* and *volume* of the frustum. The results of this experiment are in Figures 4.11 and 4.12.

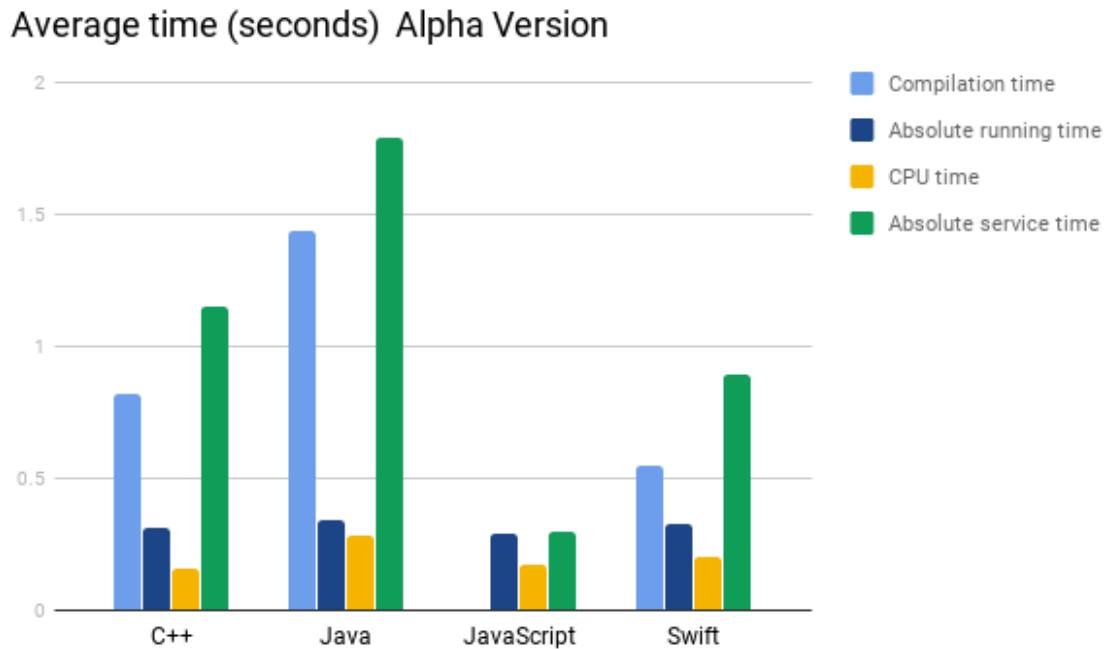


Figure 4.11: Code Speed Comparison

⁶ <http://rextester.com/>

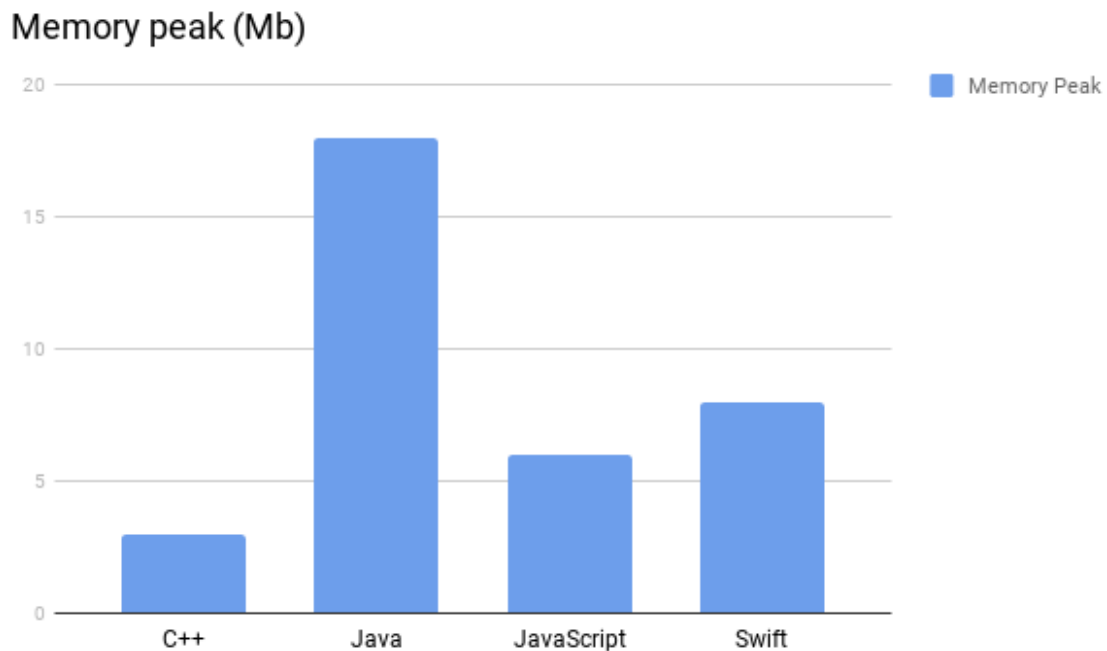


Figure 4.12: Code Memory Comparison

The two figures show that *Java* was the most time and memory intensive. *JavaScript* was the most efficient. *Swift* and *C++*, used for *ARKit* and *ARCore*, are close in time but with *C++* using the least amount of memory.

The final aspect of *Stage 1* is a determination of the number of lines of code required for each program. Table 4.1 shows the number of lines for each program.

Stage 2: Static Analysis

When static analysis was run on the different source codes. *C++* and *JavaScript* required changes while *Java* and *Swift* remained the same. *C++* only required one change to the code which was to alter the return value in the main function.

Moving back towards the competition between *ARCore* and *ARKit* for development, *C++* and *Swift* were analyzed side by side. Similar to *Java*, *C++* appeared to have no issues after static analysis. *Swift* on the other hand proved difficult to analyze as there was no clear analysis tool which provided accurate results.

Language	Number of lines
C++	67
Java	59
JavaScript	38
Script	49

Table 4.1: Number of lines - Alpha Version

Stage 3: Beta Versions

After applying all of the requested changes to the programs the speed and memory tests were run again and the results provided are shown in Figure 4.13.

Average time (seconds) - Beta Version

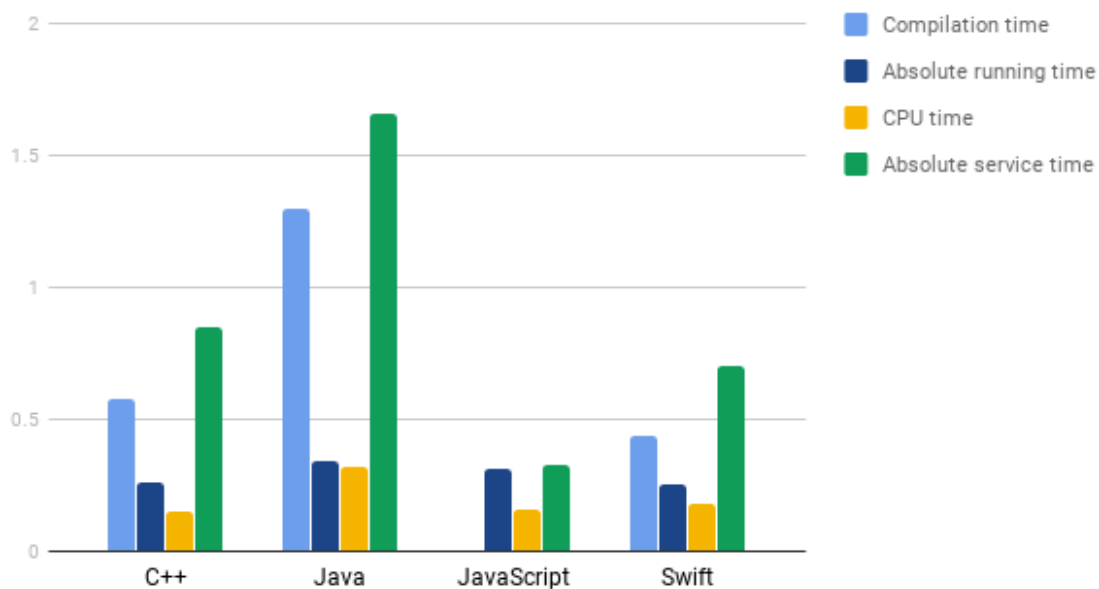


Figure 4.13: Beta Version Speed Comparison

The results showed that there was no significant difference between the speed of the *Alpha* and *Beta* versions. Even though there was a small difference between the

different iterations of the programs. However, it was not large enough to distinguish or in the case of *JavaScript* worth the time taken to make the changes. The memory consumption of the programs remained the same. Table 4.4 displays the number of lines in the beta versions of the programs. The number of lines of code for the beta versions are in Table 4.2.

Language	Number of lines
C++	68
Java	59
JavaScript	46
Script	49

Table 4.2: Number of lines - Beta Version

4.3.3 Accuracy Experiment

Stage 1: Marker-based

Table 4.3 shows the size of the markers used for this experiment in both pixels and inches. The difference in size between the two is also presented. This difference is because a *QR code* must be a perfect square. There was a size difference of 0.15% between the two types of marker.

AR Marker		
	Pixels	Inches
S	306x264	3.19x2.75
M	611x529	6.36x5.51
QR Code		
S	305x305	3.18x3.18
M	610x610	6.35x6.35

Table 4.3: Marker sizes

The dimensions of an A4 sheet of paper are $8.27 \times 11.69 \text{ in.}$ Any markers above size M would need to be printed on multiple pages. This is not possible as printers are not able to split markers and just print what the fit working from the centre of the image. This results in the sides of the marker being cut off.

Figure 4.15 presents the results of the comparison between the two types of markers. The different columns represent the minimum distance, maximum distance and average distance required to view the virtual content after the marker has been scanned. The most important measurement was the average distance because with the minimum and maximum there is the potential that measurements are not exactly correct and the average accounts for this. Figure 4.16 is a graphical representation of this experiment. It is important to note with the minimum size of the *QR code* that they are dependent on the camera being able to see all four sides of the code. This means that this distance is dependent on the size of the lens used and its corresponding *FoV*.

AR Marker			
	Min	Max	Av
S	3.9 in	37 in	20.45 in
M	6.3 in	104 in	55.15 in
QR Code			
	Min	Max	Av
S	5.9 in	31 in	18.45 in
M	7.9 in	137 in	72.45 in

Table 4.4: Marker distances

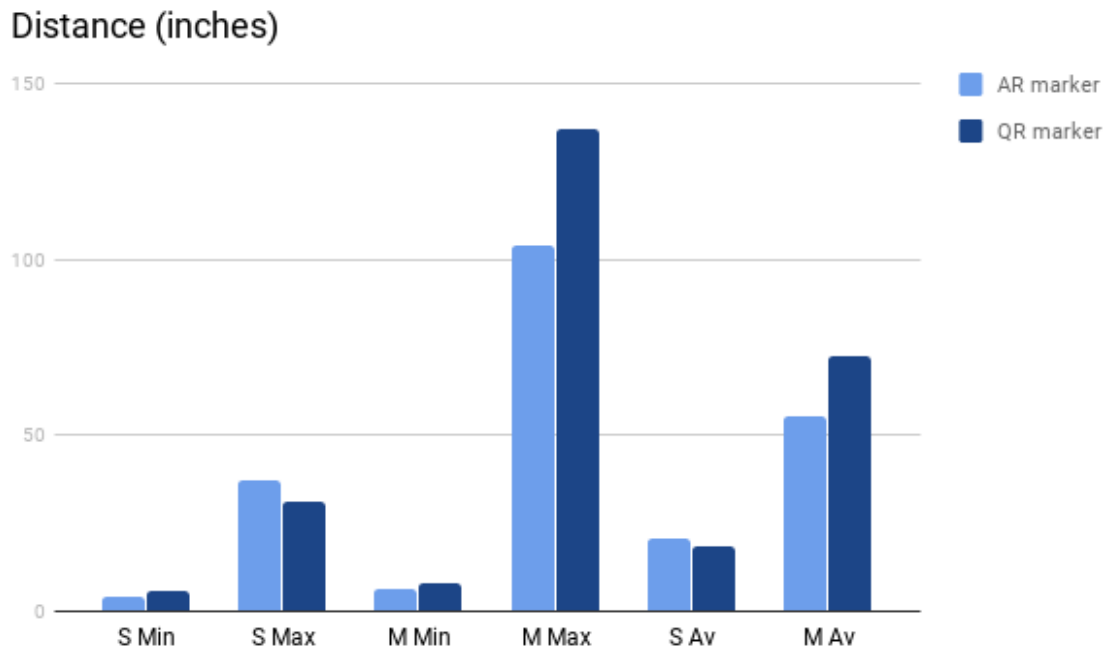


Figure 4.14: Marker comparison

Level of Interference was taken into account when the markers were scanned. In this study the interference means the amount of the marker that can be concealed while still retaining the virtual content. From scanning the markers it was revealed that AR

markers had no tolerance for interference. On the other hand *QR code* allowed for full coverage of the marker once initially scanning had been completed.

Stage 2: Markerless

For the markerless stage of this experiment the element that was accessed was the accuracy of scanned objects to become virtual elements. The two applications up for consideration were *Qlone* and *SCANN3D*.

Figures 4.17, 4.18 and 4.19 were the reference items for this experiment. Each item was placed on the *Qlone mat* and scanned.



Figure 4.15: glasses case



Figure 4.16: container



Figure 4.17: plastic cup

These experiments were unsuccessful with the applications unable to correctly scan the objects. The reasoning behind this failure is discussed in the *Analysis, Evaluation and Discussion* chapter.

4.4 Chapter Summary

This chapter started with the a description of the software used during the course of the research. Following this was a description of the marker based high level prototype and the markerless proof of concept.

The results of the experiments were presented. The order of these experiments were

UX, *Code* and *Accuracy*. There were two stages in each except for the *Code* experiment which had three.

The markerless experiment required an analysis of images to determine whether they were similar to the original.

Chapter 5

Analysis, Evaluation and Discussion

5.1 Introduction

This chapter was a critical analysis of the information gathered from the design and experimentation process in *Chapters 3 and 4*. The chapter started with a breakdown of the experimentation. This section was in two parts, how to set up the experiments and how to run them. The *UX*, *Code* and *Accuracy* experiments were individually analysed.

Following on from this there was an overall evaluation to determine if the chosen technology solved the problem at hand. Justifications regarding the change of research questions were presented.

At the end of this chapter the strengths and limitations of the study were presented and evaluated. Within this section, the problems encountered during the course of the research were addressed.

5.2 Experimentation

5.2.1 Preparing the experiment

User Experience Experiment

Stage 1: Filmmaker Surveys

In preparation for this stage the first step was to create a dataset. This dataset contained people who worked within the Irish film industry. There were two reasons for the selection of the Irish film industry aside from the location. The reasons are familiarity and the size of the industry. The familiarity came from the researcher's past working experience within the industry.

The contact details of the survey participants were obtained from the *IFTN* crew database¹. The original focus of this research project was on *personas* as a means of designing the application. *Personas* are representations of a users characteristics and work roles. (?, ?) For this reason the job titles of those contacted were *Director* and *Assistant Director*, *Director of Photography*, *Location Manager* and *Art Department*. The criteria used for selection was that the participants had a direct influence on what appeared on camera. As a result of this criteria this eliminated those who work in the lighting department.

Stage 2: Expert Interviews

Interviews were conducted to get detailed reactions from filmmakers currently working within the industry. This was a more detailed analysis and used different people than those used in *Stage 1*. The filmmakers were asked for their unbiased opinion on the research topic. These opinions were incorporated as part of the qualitative analysis of the research topic.

Initially only filmmakers were included for this stage of the experiment. This left

¹<http://www.iftn.ie/crew/>

a notable absence of a technical perspective. To alleviate these issues, experts from the Irish AR and VR industry were invited to participate. There are three organisations in Ireland. They were *VRAI* based in Dublin, *Simvirtua* from Limerick and *3MS* which operates out of Athlone, Co Westmeath. These experts provided some context to the research study as a whole. They also determined the feasibility of the application idea based on the current workflows available to developers within the industry.

Code Experiment

To prepare for this experiment, six functions were created. The first step was to split the group of functions into three.

1. **aOV() & fOV():** These functions calculated the *AoV* and *FoV* of a camera. The parameters for *aOV()* were the sensor width and focal length of the camera. The result from *aOV()* became the parameter for *fOV()* coupled with the distance to the object.
2. **latFrust(), areaFrust() & volFrust():** These functions concerned the mathematical formulas required to create the 3D image. They were lateral surface area, surface area and volume. The required parameters were the result of *fOV()*, radius of a lens and the focal length of the camera.
3. **calValues():** This final function called the previous five and returned the results to the screen.

The four programming languages chosen were *C++*, *Java*, *JavaScript*, and *Swift*. These languages were chosen for the research study due to their prominence in the creation of AR applications. In AR development some features are *SDK* dependent. The previously defined functions allowed for testing the capabilities of the languages. The majority of the preparation occurred in *Stage 1* while *Stages 2 & 3* addressed during the running of the experiment.

Accuracy Experiment

Stage 1: Marker experiment

Stage 1 of this experiment dealt with the marker based implementation of AR. To prepare for this section examples of *AR markers* and *QR codes* were sourced. Two versions of each marker were created. The markers were labelled *S* and *M* to represent small and medium. The original parameters for this experiment concerned location and lighting sources. This would have involved running the experiment indoors, outdoors and through windows. This component of the element changed in favour of an analysis of the two types of marker.

The sizes chosen for AR markers were $306 \times 265px$ and $611 \times 529px$. These were chosen after an analysis of *DPI* and the conversion of pixels to inches. The size of the *QR codes* were $305 \times 305px$ and $610 \times 610px$. There was a 0.15% size difference between the *AR markers* and *QR codes*. This was because the QR code must be a perfect square. Figure 5.1 was the *AR marker* and Figure 5.2 was the *QR code* used for this experiment.



Figure 5.1: AR Marker



Figure 5.2: QR Marker

Stage 2: Markerless Experiment

In preparation for this stage the applications *Qlone* and *SCANN3D* were downloaded. The household objects chosen were a *red plastic cup*, a *chewing gum container* and

a *glasses case*. These were chosen as a result of their differing shapes. This was to determine how the software handled objects of different sizes and shapes.

5.2.2 Running the experiment

User Experience Experiment

Stage 1: Filmmakers Survey

The response rate for this survey was in line with the response rate for surveys used in marketing campaigns. According to *Smart Insights*² the average marketing *response rate* is 4.43%. The *response rate* achieved in this study was 5.5%. It must be noted however that this study had a limit of 160 individuals while advertisers are contacting thousands.

The emails were sent in April 2018 with the recipients given a month to respond. This time limit encouraged those interested in replying to respond as soon as possible. It is important to note that the *IFTN* crew database was the source of data. This database gave direct access to individuals with the required relevant job titles. Alternatives to the use of *IFTN* were analysed but unable to yield the same results.

Stage 2: Expert interviews

There was an even split between interviews conducted in person and those done via email. This was due to the location and availability of the experts. At the time of writing two of the experts were in Spain and Limerick respectively and therefore unavailable to meet in person.

Regardless of the medium of the interview the same questions were asked depending on the field of expertise of the expert. The filmmakers and the AR experts each were asked a different sets of questions. The questions for the AR experts were on

²<https://www.smartinsights.com/email-marketing/email-communications-strategy/statistics-sources-for-email-marketing/>

a more granular level concerning the feasibility of the application and the technology used. The questions for the filmmakers focused on their understanding of the application and how it could help them.

The results from these interviews were positive. Both of the AR experts agreed with the feasibility of the idea and even referenced its marketability. The filmmakers made suggestions for features which could enhance the application's functionality. Future designs of this application would reflect the requested level of complexity.

The interviews that took place in person allowed for an extra level of discussion in relation to each answer. From one of the discussions it was discovered that the alignment of the *FoV* of the mobile device would not align with the *FoV* of the camera unless directly behind it. After further explanation it was agreed that the proposed mathematical formulas could in fact alleviate this problem.

One common thread throughout the expert interviews was the concept of marketability. As part of this study three of the four experts interviewed explicitly mentioned marketability. There were two different business models proposed depending on the industry discussed.

1. The rental of the equipment to film production companies. The cost of the devices in comparison with other filmmaking costs would need to be taken into account.
2. A *freemium* model similar to the one used for *Pokémon Go*. The idea would be that the application as standard would scan a room and display what the camera could see. However, there would only be one or two lenses available. Once the user signs up for a further subscription package there would be more lenses available.

Code Experiment

Stage 1: Alpha Version

When this stage was initially run the programming languages were basic in nature with straightforward equations and print statements. During the course of the experiment these algorithms grew in complexity. Of the six previously mentioned functions only *calValues()* was called. Using *Rextester* the values for *compilation time*, *absolute running time*, *cpu time*, *memory peak* and *absolute service time* were gathered. For consistency all tests took place on an *Acer Aspire A315-21 with Windows 10 and AMD A9-9420 processor*. During tests of the compiler platform it was discovered that the values differed each time the program was compiled. To counteract this each program was compiled three times and the average calculated. Another observation was that the *JavaScript* program required *compilation time*. A review of the source code of the compiler revealed that the website was written in *JavaScript*. This eliminated the need for any external compilation time.

Stage 2: Static Analysis

For static analysis the programming languages were analysed using the relevant software package. Of the languages *JavaScript* had the most changes required. In comparison *Java* passed the static analysis test without any issues. For *Swift* the test needed to be run on another computer. The computer chosen was *Macbook Pro with MacOS High Sierra and 2.5 GHz Intel Core i5*. *Swift* static analysers are not available for Windows machines. *Swift* is only used for programming *iOS* applications and is an *iOS* offshoot of *C#*.

Static analysis is syntax dependent. If a variable was undeclared or unused a static analyser will flag it. In a normal compiler the program will run as expected with these errors in place. The source codes before and after analysis can be seen in *Appendix A*. In some cases there was little or no difference while others had changes on almost every line.

Stage 3: Beta Version

The algorithms were updated to reflect the requested changes from *Stage 2*. The languages were put through the compiler again. Similarly to *Stage 1* each program was compiled three times. The average of the time and memory was also calculated. From the results gathered there was no significant difference between the compilation values. Even when the programs were compiled multiple times there was a change in the overall times but the average between the *alpha* and *beta* versions remained the same. This was because static analysis is only concerned with syntax and has little effect on how a program was run, and is more with its readability by others.

Accuracy Experiment

Stage 1: Marker-based

To run this experiment two copies of the markers were printed. One of each type *S* and *M*. For this experiment *AR.js* was used in conjunction with the application *3DQR* to scan the markers. While running the experiment the printed markers were placed against a wall or other surface. Holding the device the researcher started moving backwards from the marker until the virtual image appeared. This distance was the minimum distance of the marker. After recording the information, further movement backwards occurred until the model disappeared. This second distance established the maximum distance of the marker. Average distances were used due to the level of human error involved with manual measurement. The average distance accounts for human error during experimentation.

Even though the research has shown that *QR codes* are better than *AR markers* for marker-based AR, an analysis of both was conducted. Tools such as *AR.js* only operate with traditional *AR markers*. The feasibility of using *AR markers* was tested in this experiment and the overall result was the use of *AR markers* was unsuccessful. This failure was because of the following:

- The virtual content for *AR markers* disappeared if anything crossed the path between the camera and marker. This rendered the markers useless for a film

set.

- The distance of the camera from the marker led to a degradation of the image. With a *QR code* the image retained clarity until it disappeared.

Stage 2: Markerless

To run this experiment the three objects (plastic cup, glasses case and chewing gum box) were photographed using the camera on a *Samsung Galaxy J3(2016)*. Following this the objects were scanned using *Qlone* and *SCANN3D*. This experiment was to use standard applications from the Android store which could be used to complete the task of scanning objects. These two applications were unable to complete the task assigned to them for this experiment.

The errors for each application are:

- **Qlone:** This application would not read the object when it was placed on the *Qlone mat*. Despite multiple rotations and object changes it failed to render any images.
- **SCANN3D:** Using the guide the requisite number of images were scanned. For this experiment that number was *20*. With the images captured, the rendering process began. In this case, the rendering process actually caused the device to crash.

An alternative method for scanning an object to render a 3D model would be to photograph the object from multiple angles and stitch together the images using an animation program, such as *Unity* or *Blender*. Using this method would have defeated the purpose of this experiment as specialist knowledge is required.

5.3 Evaluation

This section is an evaluation of the UX framework and the experiments carried out for this research study.

The UX framework in this study aligned with current research. Further experimentation could be done using the framework but only after the creation of operational prototypes. This framework could be further manipulated depending on the type of device used. Different amounts of information are required depending on the size of the *FoV*.

Time management and the availability of suitable hardware were two problems encountered during the course of this research study. In experiments, such as the attempted sentiment analysis, discussed in *Chapter 3*, the time taken to research and write the programs greatly outweighed the significance of the results obtained. In this instance a deeper analysis of tweets could have led to the same results.

In an effort to accommodate the complexity of the *SDKs* a lot of time was spent on device modification. In the end these modifications were unsuccessful due to the built-in camera specification needed for the devices.

For future experimentation the datasets collected would need to be larger in size. A larger group would provide a broader understanding of the subject matter due to the varying experiences of its members.

5.3.1 Comparison, conclusion and suggestions

Marker-based and markerless AR were compared as part of this research study. The research conducted favoured markerless AR. One of the AR experts interviewed suggested using a combination of marker-based and markerless AR. This combination would not be successful because of the limitations of marker-based AR. This limitation concerns the marker placement in relation to the camera's position. To get the most accurate representation of a camera's *FoV* the marker would need to be placed directly on the lens. This would render the marker useless for the task. Possible solutions include an exaggeration of the *FoV* to account for the distance from the lens. However, this would greatly decrease the accuracy.

This research focused on the use of handheld mobile devices for AR. A logical next step would be the use of wearables, in particular smart glasses. The current market trend is towards the use of *Microsoft HoloLens* devices. An analysis of the cost of these devices would be required before widespread adoption could take place.

Cheaper alternatives which could be used are devices such as *Google Cardboard*. *Google Cardboard* is a housing for a mobile device made using cardboard and can be worn as glasses. The two limitations to the use of *Google Cardboard* are its weather dependent and depends on the device being used.

5.3.2 Strengths and limitations

Strengths

The strengths identified during the course of this study were:

- Pre-existing mental models
 - *Mental models* are the way a user understands how a device should work. (? , ?). From a user perspective AR is relatively easy to understand. In this instance users who are *digital natives* are accustomed to the idea of taking a photo and applying a filter. The addition of virtual information was not beyond comprehension. For *digital nomads* an understanding of AR comes from operating a camera which is “*point and click*”
- Size of hardware
 - Another strength of AR was the size of the hardware used. It was designed for handheld devices or *HMDs*. All the screen sizes used are of a relatively small size.

Limitations

The limitations that needed to be addressed are as follows:

- Technical requirements
 - This was because of the internal mechanisms required to operate AR correctly. For *ARCore*, the latest version of *Android 7.0 Nougat* is required. With *ARKit* it can only be run on the *iPhone 6* or later. This requirement rendered older devices obsolete for AR. Earlier iterations of *SDKs* such as *Google Tango* needed external hardware to operate whereas *ARCore*, *ARKit*, and *ARToolKit* use the in device capabilities. AR applications such as *Augment* operate on older devices for some tasks. There should be the ability for backwards compatibility with the *SDKs*. This could limit the scope of use instead of blocking access.
- The evolution of AR technology.
 - There are constant updates being made to AR technology as the research continues. During this research study there were several times when a concept was available for use but, as a result of the latest update became unavailable. Improvements to the technology are being made on a constant basis as often as once a month. A level of competition plays into this. When *Apple* releases their latest version of AR, *Google* are quick to follow and vice versa.
- Geometrical complexity
 - It was discovered that the geometrical calculations required to run the proposed AR application are too complex for mobile devices. This was because the calculations are too memory intensive to operate and could cause the device to crash.

5.4 Chapter Summary

This chapter began with an analysis of the experimentation conducted during this research study. The experiments for UX, code and accuracy were addressed during their preparation and execution. This was followed by an evaluation of the whole research study.

The chapter concluded with a list of the strengths and limitations of the research study. These included the mental models of the users as well as the technical limitations of AR.

Chapter 6

Conclusion

6.1 Research Overview

This research took the problem of blocking a filmed scene and applied the use of AR to solving it. *Blocking* is the placement of props and actors (either principal or background) in front of a camera. If the wrong visual information was presented it could completely change the end result of the film.

6.2 Problem Definition

The problem defined during this research project was whether AR could be used to increase the efficiency of blocking a filmed scene. The research question which was presented was:

“Can the use of a Mobile Augmented Reality (MAR) application created using marker or markerless based Augmented Reality (AR) increase the efficiency of camera and subject placement in a filmed scene?”

6.3 Design/Experimentation, Evaluation & Results

The design component of this research study involved the creation of a marker-based prototype and a markerless proof of concept. These were created using *AR.js* and *Augment* respectively.

Experiments were conducted in relation to the *UX*, *Code* and *Accuracy* of the technology with varying results.

The *UX* and *Code* experiments concluded that a markerless *MAR* written in *C++* or *Swift* would be the best suited for alleviating the proposed research problem.

6.4 Contributions and Impact

The contributions gathered from this research include the following:

1. A solution to the research problem
 - This study showed that markerless AR could be used for blocking a filmed scene. However, the cost involved in the process would be too high for some productions. This is due to the hardware and software requirements to run AR efficiently.
2. UX Framework
 - This framework provided a basis for future research in AR and could be adapted depending on the device being used.
3. Geometrical calculations
 - The geometrical calculations which were proposed would alleviate the issue of aligning the *FoV* of the camera and device. However, the calculations proved to be too memory intensive for use on smartphones.
4. Locations for use

- There are situations where the proposed application would not be worth the resources required to operate it such as filming in a small location. In a small location everyone can see the camera feed on the monitors. Some larger locations with a lot of moving parts would be better suited to the application.

6.5 Future Work & recommendations

There are several avenues for future research which include:

1. The use of machine learning for camera placement.
 - The definition of machine learning is “*an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed*”¹. When a location is scanned the AI could determine any obstacles and assign cameras accordingly. The AI would be required to follow a set of rules which could include visibility of objects, collision with surfaces, lighting sources and director preferences. Research has shown that the use of machine learning is an expanding consideration within the film industry.(?, ?) This could also work in terms of pre-production when a location is being scouted. Scouting is when a potential location is inspected for its viability as a film set. Suitability for the story of the film is also considered. During this process the director and producers could be shown all possible camera placements.
2. A further evaluation of previsualization techniques using AR.
 - Virtual objects could be incorporated in the snapshots taken for blocking the scene. These assets could include *weather conditions (clouds or mist), furniture, CGI characters (monsters) and even people*. The addition of this content would ease the transition between production and post-production.

¹<https://www.expertsystem.com/machine-learning-definition/>

3. The choice of lighting placement in a scene with the aid of AR.
 - This would use equations to calculate the levels of luminance and illuminance of lights. The same principles as the proposed application apply with the user able to determine visibility in an area.

6.6 Chapter Summary

This final chapter began with an overview of the research followed by a definition of the problem. This definition included the research question which was to be considered.

Conclusion from the design and experiments were presented next. Finally the contributions, impact and future work were discussed. The future work section talked about the incorporation of machine learning and the use of the application for lighting.

References

- Akçayır, M., Akçayır, G., Pektaş, H. M., & Ocak, M. A. (2016). Augmented reality in science laboratories: The effects of augmented reality on university students' laboratory skills and attitudes toward science laboratories. *Computers in Human Behavior*, 57, 334–342. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0747563215303253> doi: 10.1016/j.chb.2015.12.054
- Arth, C., Grasset, R., Gruber, L., Langlotz, T., Mulloni, A., & Wagner, D. (2015). The History of Mobile Augmented Reality. (May). Retrieved from <http://arxiv.org/abs/1505.01319> doi: 10.1109/INISTA.2015.7276727
- Aultman, A. (2018). Design Heuristics for Mobile Augmented Reality Game User. In *Extended abstracts of the 2018 chi conference on human factors in computing systems* (pp. 1–5). Retrieved from <https://dl.acm.org/citation.cfm?id=3188580> doi: 10.1145/3170427.3188580
- Bal, E., & Bicen, H. (2016). Computer Hardware Course Application through Augmented Reality and QR Code Integration: Achievement Levels and Views of Students. *Procedia Computer Science*, 102(August), 267–272. Retrieved from <http://dx.doi.org/10.1016/j.procs.2016.09.400> doi: 10.1016/j.procs.2016.09.400
- Beaver, F. E. (2006). *Dictionary of Film Terms: The Aesthetic Companion to Film Art*. Peter Lang. Retrieved from <https://www.peterlang.com/view/9781454194248/03{ }Copyright.html> doi: 10.3726/978-1-4539-1481-6
- Chang, D., Dooley, L., & Tuovinen, J. E. (2002). Gestalt Theory in Visual Screen Design — A New Look at an old subject. *Proceed-*

REFERENCES

- ings of the Seventh world conference on computers in education conference on Computers in education: Australian topics*, 8, 5–12. Retrieved from http://delivery.acm.org/10.1145/830000/820062/p5-chang.pdf?ip=109.76.26.39&id=820062&acc=PUBLIC&key=846C3111CE4A4710.65135ABBF91A4244.4D4702B0C3E38B35.4D4702B0C3E38B35&{_}{_}acm{_}{_}=1535704320{_}fa5f129a7358fbaf63fd12749f92c989
- Dass, N., Kim, J., Ford, S., Agarwal, S., & Chau, D. H. (2018). Augmenting Coding : Augmented Reality for Learning Programming. In *Proceedings of the sixth international symposium of chinese chi* (pp. 156–159). ACM New York, NY, USA. Retrieved from <https://dl.acm.org/citation.cfm?id=3202695>
- Digital Manufacturing and Design Innovation Institute. (2016). Augmented Reality Software & Content Generation Tools Functional Requirements for Industrial Industry Use Cases File. , 1–12. Retrieved from <http://thearea.org/ar-functional-requirements/>
- Frank, J. A., & Kapila, V. (2017). Mixed-reality learning environments: Integrating mobile interfaces with laboratory test-beds. *Computers and Education*, 110, 88–104. Retrieved from <http://dx.doi.org/10.1016/j.compedu.2017.02.009> doi: 10.1016/j.compedu.2017.02.009
- Huang, T. C., Chen, C. C., & Chou, Y. W. (2016). Animating eco-education: To see, feel, and discover in an augmented reality-based experiential learning environment. *Computers and Education*, 96, 72–82. Retrieved from <http://dx.doi.org/10.1016/j.compedu.2016.02.008> doi: 10.1016/j.compedu.2016.02.008
- Ichikari, R., Kawano, K., Kimura, A., Shibata, F., & Tamura, H. (2007). Mixed reality pre-visualization and camera-work authoring in filmmaking. In *Ismar 2006: Fifth ieee and acm international symposium on mixed and augmented reality* (pp. 239–240). Retrieved from <https://ieeexplore.ieee.org/abstract/document/4079285/> doi: 10.1109/ISMAR.2006.297823

REFERENCES

- Irshad, S., & Rambli, D. R. A. (2015). Preliminary user experience framework for designing mobile augmented reality technologies. In *2015 4th international conference on interactive digital media (icidm)* (pp. 1–4). Bandung, Indonesia: IEEE. Retrieved from <https://ieeexplore.ieee.org/document/7547833/> doi: 10.1109/IDM.2015.7547833
- Jain, D., Chinh, B., Findlater, L., Kushalnagar, R., & Froehlich, J. (2018). Exploring Augmented Reality Approaches to Real-Time Captioning: A Preliminary Autoethnographic Study. In *Proceedings of the 2018 acm conference companion publication on designing interactive systems* (pp. 7–11). Hong Kong, China: ACM New York, NY, USA. Retrieved from <https://dl.acm.org/citation.cfm?id=3205404> doi: 10.1145/3197391.3205404
- Kan, T.-W., Teng, C.-H., & Chou, W.-S. (2009). Applying QR code in augmented reality applications. *Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry - VRCAI '09*, 1(212), 253. Retrieved from <http://portal.acm.org/citation.cfm?doid=1670252.1670305> doi: 10.1145/1670252.1670305
- Kelly, A., Shapiro, R. B., de Halleux, J., & Ball, T. (2018). ARcadia. In *Proceedings of the 2018 chi conference on human factors in computing systems - chi '18* (pp. 1–8). Retrieved from <http://dl.acm.org/citation.cfm?doid=3173574.3173983> doi: 10.1145/3173574.3173983
- Kieras, D. E., & Bovair, S. (1984). The role of a mental model in learning to operate a device. *Cognitive Science*, 8(3), 255–273. Retrieved from https://onlinelibrary.wiley.com/doi/epdf/10.1207/s15516709cog0803{}_3 doi: 10.1016/S0364-0213(84)80003-8
- Kim, H., Gabbard, J. L., Anon, A. M., & Misu, T. (2018). Driver Behavior and Performance with Augmented Reality Pedestrian Collision Warning: An Outdoor User Study. *IEEE transactions on visualization and computer graphics*, 24(4), 1515–

REFERENCES

1524. Retrieved from <https://ieeexplore.ieee.org/document/8302393/> doi: 10.1109/TVCG.2018.2793680
- Lebeck, K., Ruth, K., Kohno, T., & Roesner, F. (2018). Arya: Operating System Support for Securely Augmenting Reality. *IEEE Security and Privacy*, 16(1), 44–53. Retrieved from <https://ieeexplore.ieee.org/document/8283492/> doi: 10.1109/MSP.2018.1331020
- Lino, C., Christie, M., Ranon, R., & Bares, W. (2011). The director’s lens: an intelligent assistant for virtual cinematography. In *Proceedings of the 19th international conference on multimedia 2011* (pp. 323–332). Scottsdale, AZ, USA. Retrieved from <http://dl.acm.org/citation.cfm?id=2072298.2072341> doi: 10.1145/2072298.2072341
- Liu, R., Lin, R., & Chen, S. (2018). Instant SLAM Initialization for Outdoor Omnidirectional Augmented Reality. In *Proceedings of the 31st international conference on computer animation and social agents* (pp. 66–70). Beijing, China: ACM New York, NY, USA. Retrieved from <https://dl.acm.org/citation.cfm?id=3205359> doi: 10.1145/3205326.3205359
- Liu, W., Liu, Y., & Huang, H. (2018). A Mobile Augmented Reality System For Illumination Consistency With Fisheye Camera Based On Client-Server Model. In *Casa 2018 proceedings of the 31st international conference on computer animation and social agents* (pp. 86–89). Beijing, China: ACM New York, NY, USA. Retrieved from <https://dl.acm.org/citation.cfm?doid=3205326.3205363> doi: 10.1145/3205326.3205363
- Martín-Gutiérrez, J., Fabiani, P., Benesova, W., Meneses, M. D., & Mora, C. E. (2015). Augmented reality to promote collaborative and autonomous learning in higher education. *Computers in Human Behavior*, 51, 752–761. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0747563214007110> doi: 10.1016/j.chb.2014.11.093

REFERENCES

- Marzo, A., Bossavit, B., & Ardaiz, O. (2012). Interacting with multi-touch hand-held devices in augmented reality spaces. In *Proceedings of the 13th international conference on interacción persona-ordenador - interaccion '12* (pp. 1–2). ACM. Retrieved from <http://dl.acm.org/citation.cfm?doid=2379636.2379644> doi: 10.1145/2379636.2379644
- Materna, Z., Kapinus, M., Beran, V., Smrž, P., Giuliani, M., Mirnig, N., ... Tscheligi, M. (2017). Using persona, scenario, and use case to develop a human-robot augmented reality collaborative workspace. *ACM/IEEE International Conference on Human-Robot Interaction, Part F1266*, 201–202. Retrieved from <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85016398141&doi=10.1145/%2F3029798.3038366&partnerID=40&md5=b863eb49b431a21e56ef593a827682c6> doi: 10.1145/3029798.3038366
- Milgram, P., Takemura, H., Utsumi, A., & Kishino, F. (1994). Augmented Reality: A class of displays on the reality-virtuality continuum. In *Telemanipulator and telepresence technologies* (Vol. 2351, pp. 282–292). Boston, United States. Retrieved from <http://etclab.mie.utoronto.ca/publication/1994/Milgram{-}Takemura{-}SPIE1994.pdf> doi: 10.1117/12.197321
- Mulloni, A., Seichter, H., & Schmalstieg, D. (2011). User experiences with augmented reality aided navigation on phones. *2011 10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011*, 229–230. Retrieved from <https://ieeexplore.ieee.org/document/6162892/> doi: 10.1109/ISMAR.2011.6092390
- Nazri, N. I. A. M., Rambli, D. R. A., & Tomi, A. (2015). A mobile augmented reality game design approach for on product advertising. In *Proceedings of the 12th international conference on advances in computer entertainment technology - ace '15* (pp. 1–8). Retrieved from <http://dl.acm.org/citation.cfm?doid=2832932.2856222> doi: 10.1145/2832932.2856222
- Nebeling, M., Nebeling, J., Yu, A., & Rumble, R. (2018). ProtoAR : Rapid Physical-Digital Prototyping of Mobile Augmented Reality Applications. In *Proceedings of the*

REFERENCES

- 2018 chi conference on human factors in computing systems* (pp. 1–12). Montreal, QC, Canada: ACM New York, NY, USA. Retrieved from <https://dl.acm.org/citation.cfm?id=3173927> doi: 10.1145/3173574.3173927
- Nguyen, M., Yeap, W., & Hooper, S. (2017). Design of a new trading card for table-top augmented reality game environment. In *International conference image and vision computing new zealand* (pp. 1–6). Palmerston North, New Zealand: IEEE. Retrieved from <https://ieeexplore.ieee.org/document/7804458/> doi: 10.1109/IVCNZ.2016.7804458
- Nivedha, S., & Hemalatha, S. (2015). Enhancing user experience through physical interaction in handheld Augmented Reality. In *2015 international conference on computer communication and informatics, iccci 2015* (pp. 1–7). Coimbatore, India: IEEE. Retrieved from <https://ieeexplore.ieee.org/abstract/document/7218127/> doi: 10.1109/ICCCI.2015.7218127
- Norman, D. a. (1986). *Cognitive Engineering*. Retrieved from <https://pdfs.semanticscholar.org/57f1/76992f92ae559d9c110211d7f04c5143cb44.pdf>
- Olsson, T., Lagerstam, E., Kärkkäinen, T., & Väänänen-Vainio-Mattila, K. (2013). Expected user experience of mobile augmented reality services: A user study in the context of shopping centres. *Personal and Ubiquitous Computing*, 17(2), 287–304. Retrieved from <https://dl.acm.org/citation.cfm?id=2434685> doi: 10.1007/s00779-011-0494-x
- Pallot, M., Eynard, R., Poussard, B., Christmann, O., & Richir, S. (2013). Augmented sport. In *Proceedings of the virtual reality international conference on laval virtual - vric '13* (pp. 1–8). Retrieved from <http://dl.acm.org/citation.cfm?doid=2466816.2466821> doi: 10.1145/2466816.2466821
- Park, D., Nam, T.-J., & Shi, C.-K. (2006). Designing an immersive tour experience system for cultural tour sites. In *Chi '06 extended abstracts on human factors in computing systems - chi ea '06* (p. 1193). Retrieved from <https://dl.acm.org/citation.cfm?doid=1125451.1125675> doi: 10.1145/1125451.1125675

REFERENCES

- Paulo Lima, J., Roberto, R., Simões, F., Almeida, M., Figueiredo, L., Marcelo Teixeira, J., & Teichrieb, V. (2017). Markerless tracking system for augmented reality in the automotive industry. *Expert Systems with Applications*, 82, 100–114. Retrieved from <http://dx.doi.org/10.1016/j.eswa.2017.03.060> doi: 10.1016/j.eswa.2017.03.060
- Pranoto, H. (2017). 2017-ICIMTech-Usability testing method in Augmented Reality Application ver3. In *2017 international conference on information management and technology (icimtech)* (pp. 181–186). Yogyakarta, Indonesia: IEEE. Retrieved from <https://ieeexplore.ieee.org/document/8273534/> doi: 10.1109/ICIMTech.2017.8273534
- Prensky, M. (2001). Digital Natives, Digital Immigrants. *On the Horizon*, 9(5), 1–6. Retrieved from <https://www.emeraldinsight.com/doi/abs/10.1108/10748120110424816> doi: 10.1108/10748120110424816
- Quint, F., Sebastian, K., & Gorecky, D. (2015). A Mixed-reality Learning Environment. *Procedia Computer Science*, 75(Vare), 43–48. Retrieved from <http://dx.doi.org/10.1016/j.procs.2015.12.199> doi: 10.1016/j.procs.2015.12.199
- Ren, K., & Wang, C. (2012). Security Challenges for the Public Cloud. *Internet Computing, IEEE*, 16(1), 69–74. Retrieved from <http://ieeexplore.ieee.org/xpls/abs{all}.jsp?arnumber=6123700> doi: 10.1109/MIC.2012.14
- Sekhavat, Y. A. (2016). KioskAR: An Augmented Reality Game as a New Business Model to Present Artworks. *International Journal of Computer Games Technology*, 2016, 1–12. Retrieved from <https://dl.acm.org/citation.cfm?id=2985982> doi: 10.1155/2016/7690754
- Shea, R., Fu, D., Sun, A., Cai, C., Ma, X., Fan, X., . . . Liu, J. (2017). Location-Based Augmented Reality with Pervasive Smartphone Sensors: Inside and beyond Pokemon Go! *IEEE Access*, 5, 9619–9631. Retrieved from <https://ieeexplore.ieee.org/document/7907241/> doi: 10.1109/ACCESS.2017.2696953

REFERENCES

- Spielmann, S., Schuster, A., Götz, K., & Helzle, V. (2016). VPET – A Toolset for Collaborative Virtual Filmmaking. In *Siggraph asia 2016 technical briefs* (pp. 1–4). Retrieved from <https://www.dreamspaceproject.eu/dyn/1479378270265/SA16{-}VPET{-}Filmakademie.pdf> doi: 10.1145/3005358.3005370
- Sun, K., Mohta, K., Pfrommer, B., Watterson, M., Liu, S., Mulgaonkar, Y., ... Kumar, V. (2017). Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight. *IEEE Robotics and Automation Letters*, 3(2), 965 – 972. Retrieved from <http://arxiv.org/abs/1712.00036> doi: 10.1109/LRA.2018.2793349
- Touel, S., Mekkadem, M., Kenoui, M., & Benbelkacem, S. (2017). Collocated learning experience within collaborative augmented environment (anatomy course). In *2017 5th international conference on electrical engineering - boumerdes (icee-b)* (pp. 1–5). Retrieved from <http://ieeexplore.ieee.org/document/8192219/> doi: 10.1109/ICEE-B.2017.8192219
- Tsai, C. H., & Huang, J. Y. (2018). Augmented reality display based on user behavior. *Computer Standards and Interfaces*, 55(August 2017), 171–181. Retrieved from <https://doi.org/10.1016/j.csi.2017.08.003> doi: 10.1016/j.csi.2017.08.003
- Vazquez-Alvarez, Y., Aylett, M. P., Brewster, S. A., Jungenfeld, R. V., & Virolainen, A. (2015). Designing Interactions with Multilevel Auditory Displays in Mobile Audio-Augmented Reality. *ACM Transactions on Computer-Human Interaction*, 23(1), 1–30. Retrieved from <http://dl.acm.org/citation.cfm?doid=2872314.2829944> doi: 10.1145/2829944

Appendix A

Code

A.1 Experiments

A.1.1 Alpha Versions

C++

```
#include <iostream>
#include <math.h>
using namespace std;

// function declaration
double a0V(double sw, int h);
double f0V(double a, int dis);
double areaFrust(double R, double r, int h);
double latFrust(double R, double r, int dis);
double volFrust(double R, double r, int h);
double calValues(int h, int dis, double r, double sw);

int main () {
int h = 24; //focal length
int dis = 13; //distance also L
```

```
double r = 7.5; //radius
double sw = 15.5; //sensor width
double cal;

    cal = calValues(h, dis, r, sw);

    return 0;
}

//aOV = 2 ARCTAN(Sensor width / (2 * focal length)) * (180 / pi)
double aOV(double sw, int h) {
    double a;
    a = (2 * (atan(sw / (2 * h)) * (180 / M_PI)));
    return a;
}

//fOV = 2 (TAN (Angle of View / 2) * Distance to Subject)
double fOV(double a, int dis) {
    double R;
    R = 2 * (tan(a/2)) * dis;
    return R;
}

//latFrust = (pi*(R+r))*L
double latFrust(double R, double r, int dis) {
    double lat;
    lat = (M_PI * (R+r)) * dis;
    return lat;
}

//areaFrust = pi(R + r)* sqrt(R - r)^2+h^2
double areaFrust(double R, double r, int h) {
    double c;
    c = M_PI * (R + r) * sqrt(((R - r)*(R-r))+(h*h));
```

```
        return c;
    }
//volFrust = pi/3h(R^2+r^2+R*r)
double volFrust(double R, double r, int h) {
    double d;
    d = M_PI/ (3 * 12) * ((R*R)+(r*r)+(R*r));
    return d;
}
//Calculate all of the values
double calValues(int h, int dis, double r, double sw){
    double a;
    a = aOV(sw, h);
    cout << "The Angle of View is " << a << endl;
    double R;
    R = fOV(a, dis);
    cout << "The Linear Field of View is " << R << endl;
    double c;
    c = areaFrust(R, r, h);
    cout << "The Surface Area is " << c << endl;
    double lat;
    lat = latFrust(R, r, dis);
    cout << "The Lateral Area is" << lat << endl;
    double d;
    d = volFrust(R, r, h);
    cout << "The Volume is " << d << endl;
}
```

Java

```
import java.util.*;
import java.lang.*;
```

```
class Rextester {

    public static void main(String[] args) {

        int a = 11;
        int b = 6;
        int h = 24; //focal length
        int dis = 13; //distance also L
        double r = 7.5; //radius
        double sw = 15.5; //sensor width
        double R = 1;

        //Calling the function
        double cal = calValues(h,dis,r,sw);
        System.out.print(cal);
    }

    public static double aOV(double sw,int h) {
        double a;
        //Angle of View(in degrees) = 2 ARCTAN(Sensor width /
        (2 * focal length)) * (180 / pi)
        a = 2 * (Math.atan(sw / (2 * h))) * (180 / Math.PI);

        return a;
    }

    public static double fOV(double a, int dis) {
        double R;
        //Linear Field of View = 2 (TAN (Angle of View / 2) * Distance to Subject)
        R = 2*(Math.tan(a/2)) * dis;

        return R;
    }
}
```

```
    }

    public static double latFrust(double R, double r, int dis) {
double lat;

        //Lateral area of frustum = (pi*(R+r))*L
lat = (Math.PI * (R+r))* dis;

        return lat;
}

    public static double areaFrust(double R, double r, int h) {
        double c;

        //Surface area of frustum = pi(R + r)* sqrt(R - r)^2+h^2
        c = (Math.PI*(R+r) * (Math.sqrt((R-r)*(R-r)+(h*h))));

        return c;
}

    public static double volFrust(double R, double r, int h) {
        double d;

        //Volume of frustum = pi/3h(R^2+r^2+R*r)
        d = (Math.PI / (3*h)*((R*R)+(r*r)+(R*r)));

        return d;
}

    public static double calValues(int h, int dis, double r, double sw){
        double a = aOV(sw,h);
        System.out.println("The angle of View is " + a);
        double R = fOV(a,dis);
        System.out.println("The linear field of view is " + R);
        double c = areaFrust(R,r,h);
        System.out.println("The surface area of frustum is " + c);
        double lat = latFrust(R,r,h);
        System.out.println("The lateral area of frustum is " + lat);
    }
```

```
double d= volFrust(R,r,h);  
System.out.println("The volume of the frustum is " + d);  
return 0;  
}  
}
```

JavaScript

```
h = 24; //focal length  
dis = 13; //distance also L  
r = 7.5; //radius  
sw = 15.5; //sensor width  
  
function aOV(sw,h) {  
    //aOV = 2 ARCTAN(Sensor width / (2 * focal length)) * (180 / pi)  
    a = 2 * (Math.atan(30 /(2 * h)) * (180 / Math.PI));  
    print("The angle of view is " + a);  
}  
  
function fOV(a,dis) {  
    //fOV = 2 (TAN (Angle of View / 2) * Distance to Subject)  
    R = 2*(Math.tan(a/2)) * dis;  
    print("The Linear Field of View is " + R);  
}  
  
function latFrust(R,r,dis) {  
    //latFrust = (pi*(R+r))*L  
    lat = (Math.PI * (R+r)) * dis;  
    print("The Lateral Area of Frustum is " + lat);  
}
```



```
function areaFrust(R,r,h) {
    //areaFrust = pi(R + r)* sqrt(R - r)^2+h^2)
    c = (Math.PI * R + r) * Math.sqrt(((R - r)^2) + (h^2));
    print("The surface area of frustum " + c);
}

function volFrust(R,r,h) {
    //volFrust = pi/3h(R^2+r^2+R*r)
    d = (Math.PI / (3*h) * ((R^2)+(r^2)+(R*r)));
    print("The volume of frustum is " + d);
}

function calValues(h,dis,r,sw) {
    //Call all of the other functions
    print(aOV(sw,h));
    print(fOV(a,dis));
    print(latFrust(R,r,dis));
    print(areaFrust(R,r,h));
    print(volFrust(R,r,h));
}

calValues(h,dis,r,sw);
```

Swift

```
import Foundation

let h = 24; //focal length
let dis = 13; //also L
let r = 7.5;
let sw = 15.5;
```

```
print (calValues(h:dis:r:sw:))

func aOV(sw: Double, h: Int) -> Double {
  //aOV = 2 ARCTAN(Sensor width / (2 * focal length)) * (180 / pi)
  let a = (2*atan(sw / Double(2*h))) * (180 / Double.pi)
  return a
}

func fOV(a: Double, dis: Int) -> Double {
  //fOV = 2 (TAN (Angle of View / 2) * Distance to Subject)
  let R = 2*(tan(a / 2) * Double(dis))
  return R
}

func latFrust(R: Double, r: Double, h: Int) -> Double {
  //latFrust = (pi*(R+r))*L
  let lat = (Double.pi * Double(R+r)) * Double(dis)
  return lat
}

func areaFrust(R: Double, r: Double, h: Int) -> Double {
  //areaFrust = pi(R + r)* sqrt((R - r)^2+h^2)
  let x = R-r;
  let c = (Double(M_PI)*(R + r)) * (((Double(x*x) + Double(h*h)).squareRoot()))
  return c
}

func volFrust(R: Double, r: Double, h: Int) -> Double {
  //volFrust = pi/3h(R^2+r^2+R*r)
  let y = R*R;
  let z = r*r;
  let d = (Double(M_PI) / Double(3 * h))
  let e = (y + z + (R*r))
  let f = Double(d) / Double(e)
```

```
return f
}

func calValues(h: Int, dis: Int, r: Double, double sw: Double) -> Double {

    let a = aOV(sw:h:)
    print("The angle of view is ",a)
    let R = fOV(a:dis:)
    print ("The linear field of view is",R)
    let c = areaFrust(R:r:h:)
    print ("The surface area of frustum is", c)
    let lat = latFrust(R:r:h:)
    print ("The lateral area of frustum is",lat)
    let d = volFrust(R:r:h:)
    print ("The volume of frustum is",d)
    return 0;
}
```

A.1.2 Beta Versions

C++

```
#include <iostream>
#include <math.h>
using namespace std;

// function declaration
double aOV(double sw, int h);
double fOV(double a, int dis);
double areaFrust(double R, double r, int h);
double latFrust(double R, double r, int dis);
double volFrust(double R, double r, int h);
```

```
double calValues(int h, int dis, double r, double sw);

int main () {

    int h = 24; //focal length
    int dis = 13; //distance also L
    double r = 7.5; //radius
    double sw = 15.5; //sensor width
    double cal;

    cal = calValues(h, dis, r, sw);

    return cal;
}

//Angle of View(in degrees) = 2 ARCTAN(Sensor width / (2 * focal length)) * (180 / pi)
double aOV(double sw, int h) {
    double a;
    a = (2 * (atan(sw / (2 * h)) * (180 / M_PI)));

    return a;
}

//Linear Field of View = 2 (TAN (Angle of View / 2) * Distance to Subject)
double fOV(double a, int dis) {
    double R;
    R = 2 * (tan(a/2)) * dis;

    return R;
}
```

```
//Lateral area of frustum = (pi*(R+r))*L
double latFrust(double R, double r, int dis) {
    double lat;
    lat = (M_PI * (R+r)) * dis;

    return lat;
}

//Surface area of frustum = pi(R + r)* sqrt(R - r)^2+h^2)
double areaFrust(double R, double r, int h) {
    double c;
    c = M_PI * (R + r) * sqrt(((R - r)*(R-r))+(h*h));

    return c;
}

//Volume of frustum = pi/3h(R^2+r^2+R*r)
double volFrust(double R, double r, int h) {
    double d;
    d = M_PI/ (3 * 12) * ((R*R)+(r*r)+(R*r));

    return d;
}

//Calculate all of the values
double calValues(int h, int dis, double r, double sw){
    double a;
    a = a0V(sw, h);
    cout << "The Angle of View is " << a << endl;
```

```
double R;
R = fOV(a, dis);
cout << "The Linear Field of View is " << R << endl;
double c;
c = areaFrust(R, r, h);
cout << "The Surface Area is " << c << endl;
double lat;
lat = latFrust(R, r, dis);
cout << "The Lateral Area is" << lat << endl;
double d;
d = volFrust(R, r, h);
cout << "The Volume is " << d << endl;
}
```

Java

```
import java.util.*;
import java.lang.*;

class Rextester {

    public static void main(String[] args) {
        int a = 11;
        int b = 6;
        int h = 24; //focal length
        int dis = 13; //distance also L
        double r = 7.5; //radius
        double sw = 15.5; //sensor width
        double R = 1;

        //Calling the function
```

```
        double cal = calValues(h,dis,r,sw);
        System.out.print(cal);

    }

    public static double aOV(double sw,int h) {
        double a;
        //Angle of View(in degrees) = 2 ARCTAN(Sensor width / (2 * focal length)) * (180 / Math.PI)
        a = 2 * (Math.atan(sw / (2 * h))) * (180 / Math.PI);

        return a;
    }

    public static double fOV(double a, int dis) {
        double R;
        //Linear Field of View = 2 (TAN (Angle of View / 2) * Distance to Subject)
        R = 2*(Math.tan(a/2)) * dis;

        return R;
    }

    public static double latFrust(double R, double r, int dis) {
        double lat;
        //Lateral area of frustum = (pi*(R+r))*L
        lat = (Math.PI * (R+r))* dis;

        return lat;
    }
}
```

```
public static double areaFrust(double R, double r, int h) {
    double c;
    //Surface area of frustum =  $\pi(R + r) \cdot \sqrt{(R - r)^2 + h^2}$ 
    c = (Math.PI*(R+r) * (Math.sqrt((R-r)*(R-r)+(h*h))));

    return c;
}

public static double volFrust(double R, double r, int h) {
    double d;
    //Volume of frustum =  $\pi/3h(R^2 + r^2 + R \cdot r)$ 
    d = (Math.PI / (3*h)*((R*R)+(r*r)+(R*r)));

    return d;
}

public static double calValues(int h, int dis, double r, double sw){
    double a = aOV(sw,h);
    System.out.println("The angle of View is " + a);
    double R = fOV(a,dis);
    System.out.println("The linear field of view is " + R);
    double c = areaFrust(R,r,h);
    System.out.println("The surface area of frustum is " + c);
    double lat = latFrust(R,r,h);
    System.out.println("The lateral area of frustum is " + lat);
    double d= volFrust(R,r,h);
    System.out.println("The volume of the frustum is " + d);
    return 0;
}
```



```
}
```

JavaScript

```
<script>
var h = 24; //focal length
var dis = 13; //distance also L
var r = 7.5; //radius
var sw = 15.5; //sensor width
var a;
var R;

function aOV(sw, h) {
    "use strict";
    //AoV = 2 ARCTAN(Sensor width / (2 * focal length)) * (180 / pi)
    a = 2 * (Math.atan(sw / (2 * h)) * (180 / Math.PI));
    return ("The angle of view is " + a);
}

function fOV(a, dis) {
    "use strict";
    //FoV = 2 (TAN (Angle of View / 2) * Distance to Subject)
    R = 2 * (Math.tan(a / 2)) * dis;
    return ("The Linear Field of View is " + R);
}

function latFrust(R, r, dis) {
    "use strict";
    //Lateral area of frustum = (pi*(R + r))*L
    var lat = (Math.PI * (R + r)) * dis;
    return ("The Lateral Area of Frustum is " + lat);
}
```

```
}
```

```
function areaFrust(R, r, h) {  
    "use strict";  
    //Surface area of frustum =  $\pi(R + r) * \sqrt{(R - r)^2 + (h * h)}$   
    var c = (Math.PI * R + r) * Math.sqrt(((R - r) * (R - r)) + (h * h));  
    return ("The surface area of frustum " + c);  
}
```

```
function volFrust(R, r, h) {  
    "use strict";  
    //Volume of frustum =  $\pi/3h(R^2 + r^2 + R * r)$   
    var d = (Math.PI / (3 * h) * ((R * R) + (r * r) + (R * r)));  
    return ("The volume of frustum is " + d);  
}
```

```
function calValues(h, dis, r, sw) {  
    "use strict";  
    //Call all of the other functions  
    aOV(sw, h);  
    fOV(a, dis);  
    latFrust(R, r, dis);  
    areaFrust(R, r, h);  
    volFrust(R, r, h);  
}
```

```
calValues(h, dis, r, sw);
```

```
</script>
```

Swift

A.2 Prototypes

A.2.1 AR.js

```
<script src="https://aframe.io/releases/0.6.0/aframe.min.js"></script>
<script src="https://jeromeetienne.github.io/AR.js/aframe/build/
  aframe/build/aframe-ar.js">
<body style='margin : 0px; overflow: hidden;'>
  <a-scene embedded arjs='sourceType: webcam;'>
    <!-- handle marker with your own pattern -->
    <a-marker type='pattern' url='path/to/pattern-marker.patt'>
      <a-box position='0 0.5 0' material='color: red;'></a-box>
    </a-marker>
    <!-- handle marker with hiro preset -->
    <a-marker preset='hiro'>
      <a-box position='0 0.5 0' material='color: green;'></a-box>
    </a-marker>
    <!-- handle barcode marker -->
    <a-marker type='barcode' value='5'>
      <a-box position='0 0.5 0' material='color: blue;'></a-box>
    </a-marker>
    <!-- add a simple camera -->
    <a-entity camera></a-entity>
  </a-scene>
</body>
```

Appendix B

User Experience Questions

B.1 Survey Questions & Responses

- ***Q1. Are you familiar with AR and if so have you ever used an AR application?***
 - *“Have heard about it but not used it.”*
 - *“Familiar with use but have never used it.”*
 - *“I am familiar with it, however, I have not used it and given the type of director I am – human stories/drama etc., it probably unlikely I would do so.”*
 - *“I am familiar with what AR is, but have never used AR for a production.”*
 - *“I am not familiar with AR and have yet to use an AR app.”*
 - *“Not Really familiar and no have not used an AR application.”*
- ***Q2. Would you ever consider using an AR application for work?***
 - *“Yes, it could help the re-design of existing buildings.”*
 - *“Yes, could be useful in a creative presentation to potential financiers.”*
 - *“Unlikely, but you just never know what comes along! Never say Never!”*

- *“I would consider it if a suitable project came up and an AR App would streamline the production process. It would have to be a benefit: time, creativity or financial, and not unnecessarily complicate the process.”*
- *“I would be very interested in using AR in my workspace.”*
- *“Would need to know a lot more about it and the costs involved.”*
- ***Q3. As part of your daily work are there aspects if any which could be improved by the use of AR?***
 - *“Yes, it would be great for realtime concepts”*
 - *“Not sure?”*
 - *“Once again, no. I tell “human, emotional stories”, however...”*
 - *“AR Apps could be of use, but daily, not so sure. As I’m sure you’re aware there are AR apps that can be used to plot the suns position or block out scenes, plot camera positions, etc. I haven’t used them though. On the suns position, a compass would give enough information for most needs - where due south is, and so what path the sun takes around a location. It would have to be a very specific shot in order to need the actual azimuth. I can see how AR could potentially be useful if a production involved CGI and live action, or where detailed storyboards can be a requirement. And situations where corporate clients are involved, they tend to like everything laid out for them and often aren’t very visually literate, so supporting material produced through an app could potentially be useful.”*
 - *“It sounds like it could work well in a green screen situation where I could line up the plate shots with the green screen shots etc. So that the character can work within the eventual environs which would be built in post.”*

B.2 Expert Interviews

B.2.1 AR Experts

Q1. From an industry standpoint does the proposed idea sound feasible?

Q2. Which programming language would you recommend for the development of the application and why?

Q3. Would you have a preference over the use of marker or marker-less AR from a development point of view?

Q4. Based on your knowledge and experience have you any feedback about the proposed application?

Q5. What improvements would you make to the design of the application?

Q6. Have you any recommendations for potential areas of research within AR as a whole?

B.2.2 Filmmakers

Q1. Are you familiar with Augmented Reality?

Q2. Would you consider using an Augmented Reality application as part of your daily work?

Q3. Does the idea presented for the application make sense for its intended use?

Q4. What improvements would you make to the design of the application?

Q5. What alternatives would you see for the use of Augmented Reality applications on film sets?